DCTM: Discrete-Continuous Transformation Matching for Semantic Flow*

Seungryong Kim[1], Dongbo Min[2], Stephen Lin[3], and Kwanghoon Sohn[1]
[1]Yonsei University [2]Chungnam National University [3]Microsoft Research
{srkim89,khsohn}@yonsei.ac.kr dbmin@cnu.ac.kr stevelin@microsoft.com

Abstract

Techniques for dense semantic correspondence have provided limited ability to deal with the geometric variations that commonly exist between semantically similar images. While variations due to scale and rotation have been examined, there is a lack of practical solutions for more complex deformations such as affine transformations because of the tremendous size of the associated solution space. To address this problem, we present a discrete-continuous transformation matching (DCTM) framework where dense affine transformation fields are inferred through a discrete label optimization in which the labels are iteratively updated via continuous regularization. In this way, our approach draws solutions from the continuous space of affine transformations in a manner that can be computed efficiently through constant-time edge-aware filtering and a proposed affine-varying CNN-based descriptor. Experimental results show that this model outperforms the state-of-the-art methods for dense semantic correspondence on various benchmarks.

1. Introduction

Establishing dense correspondences across *semantically* similar images is essential for numerous tasks such as non-parametric scene parsing, scene recognition, image registration, semantic segmentation, and image editing [15, 33, 32].

Unlike traditional dense correspondence for estimating depth [46] or optical flow [9, 51], semantic correspondence estimation poses additional challenges due to intra-class appearance and shape variations among object instances, which can degrade matching by conventional approaches [33, 59]. Recently, several methods have attempted to deal with the appearance differences using convolutional neural network (CNN) based descriptors because of their high invariance to appearance variations [34, 11, 61, 24]. However, geometric variations are considered in just a limited manner through constraint settings such as those used for depth or optical flow. Some methods solve for geometric variations such as scale or rotation [18, 41, 21], but they consider only a discrete set of scales or rotations as possible solutions, and do not capture the non-rigid geometric deformations that commonly exist between semantically similar images.

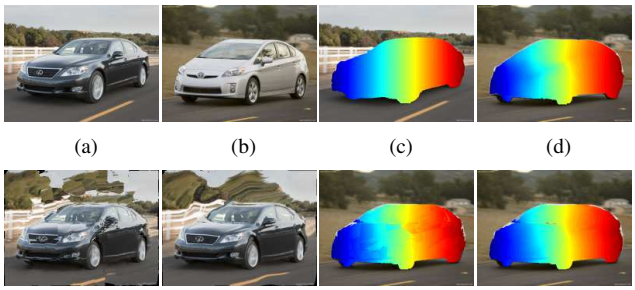


Figure 1. Visualization of our DCTM results: (a) source image, (b) target image, (c), (d) ground truth correspondences, (e), (f), (g), (h) warped images and correspondences after discrete and continuous optimization, respectively. For images undergoing non-rigid deformations, our DCTM estimates reliable correspondences by iteratively optimizing the label space via continuous regularization.

It has been shown that these non-rigid image deformations can be locally well approximated by affine transformations [45, 30, 29]. To estimate dense affine transformation fields, a possible approach is to discretize the space of affine transformations and find a labeling solution. However, the higher-dimensional search space for affine transformations makes discrete global optimization algorithms such as graph cut [6] and belief propagation [48, 52] computationally infeasible. For more efficient optimization over large label spaces, the PatchMatch Filter (PMF) [37] integrates constant-time edge-aware filtering (EAF) [43, 36] with PatchMatch-based randomized search [2]. PMF is leveraged for dense semantic correspondence in DAISY Filter Flow (DFF) [59], which finds labels for displacement fields as well as for scale and rotation. Extending DFF to affine transformations would be challenging though. One reason is that its efficient technique for computing DAISY features [54] at pre-determined scales and rotations cannot be applied for affine transformations. Another reason is that, as shown in [27, 21], the weak implicit smoothing em-

*This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No.2016-0-00197).

bedded in PMF makes it more susceptible to erroneous local minima, and this problem may be magnified in the higher-dimensional affine transformation space. Explicit smoothing models have been adopted to alleviate this problem in the context of stereo matching [28, 3], but were designed specifically for depth regularization.

In this paper, we introduce an effective method for estimating dense affine transformation fields between semantically similar images, as shown in Fig. 1. The key idea is to couple a discrete local labeling optimization with a continuous global regularization that updates the discrete candidate labels. An affine transformation field is efficiently inferred in a filter-based discrete labeling scheme inspired by PMF, and then the discrete affine transformation field is globally regularized in a moving least squares (MLS) manner [45]. These two steps are iterated in alternation until convergence. Through the synergy of the discrete local labeling and continuous global regularization, our method yields *continuous* solutions from the space of affine transformations, rather than selecting from a pre-defined, finite set of discrete samples. We show that this continuous regularization additionally overcomes the aforementioned implicit smoothness problem in PMF.

Moreover, we model the effects of affine transformations directly within the state-of-the-art fully convolutional self-similarity (FCSS) descriptor [24], which leads to significant improvements in processing speed over computing descriptors on various affine transformations of the image. Experimental results show that the presented model outperforms the latest methods for dense semantic correspondence on several benchmarks, including that of Taniai *et al.* [53], Proposal Flow [16], and PASCAL [10].

2. Related Work

Dense Semantic Flow Most conventional techniques for dense semantic correspondence have employed handcrafted features such as SIFT [35] or DAISY [54]. To improve matching quality, they have focused on optimization. Liu *et al.* [33] pioneered the idea of dense correspondence across different scenes, and proposed SIFT Flow which is based on hierarchical dual-layer belief propagation. Inspired by this, Kim *et al.* [23] proposed the deformable spatial pyramid (DSP) which performs multi-scale regularization with a hierarchical graph. Among other methods are those that take an exemplar-LDA approach [7], employ joint image set alignment [62], or jointly solve for cosegmentation [53].

Recently, CNN-based descriptors have been used to establish dense semantic correspondences. Zhou *et al.* [61] proposed a deep network that exploits cycle-consistency with a 3D CAD model [40] as a supervisory signal. Choy *et al.* [11] proposed the universal correspondence network (UCN) based on fully convolutional feature learning. Most recently, Kim *et al.* [24] proposed the FCSS descriptor that formulates local self-similarity (LSS) [47] within a fully convolutional network. Because of its LSS-based structure, FCSS is inherently insensitive to intra-class appearance variations while maintaining precise localization ability. However, none of these methods is able to handle non-rigid geometric variations.

Several methods aim to alleviate geometric variations through extensions of SIFT Flow, including scale-less SIFT Flow (SLS) [18], scale-space SIFT Flow (SSF) [41], and generalized DSP (GDSP) [21]. However, these techniques have a critical practical limitation that their computation increases linearly with the search space size. A generalized PatchMatch algorithm [2] was proposed for efficient matching that leverages a randomized search scheme. This was utilized by HaCohen *et al.* [15] in a non-rigid dense correspondence (NRDC) algorithm, but employs weak matching evidence that cannot guarantee reliable performance. Geometric invariance to scale and rotation is provided by DFF [59], but its implicit smoothing model which relies on randomized sampling and propagation of good estimates in the direct neighborhood often induces mismatches. A segmentation-aware approach [56] was proposed to provide geometric robustness for descriptors, but can have a negative effect on the discriminative power of the descriptor. Recently, Ham *et al.* [16] presented the Proposal Flow (PF) algorithm to estimate correspondences using object proposals. While these aforementioned techniques provide some amount of geometric invariance, none of them can deal with affine transformations across images, which are a frequent occurrence in dense semantic correspondence.

Image Manipulation A possible approach for estimating dense affine transformation fields is to interpolate sparsely matched points using a method, including thin plate splines (TPS) [4], motion coherence [60], coherence point drift [39], or smoothly varying affine stitching [30]. MLS is also a scattered point interpolation technique first introduced in [26] to reconstruct a continuous function from a set of point samples by incorporating spatially-weighted least squares. MLS has been successfully used in applications such as image deformation [45], surface reconstruction [13], image super-resolution and denoising [5], and color transfer [22]. Inspired by the MLS concept, our method utilizes it to regularize estimated affine fields, but with a different weight function and an efficient computational scheme.

More related to our work is the method of Lin *et al.* [29], which jointly estimates correspondence and relative patch orientation for descriptors. However, it is formulated with pre-computed sparse correspondences and also requires considerable computation to solve a complex non-linear optimization. By contrast, our method adopts dense descriptors that can be evaluated efficiently for any affine transformation, and employs quadratic continuous optimization to rapidly infer dense affine transformation fields.

3. Method

3.1. Problem Formulation and Model

Given a pair of images I and I', the objective of dense correspondence estimation is to establish a correspondence i' for each pixel $i = [i_{\mathbf{x}}, i_{\mathbf{y}}]$. Unlike conventional dense correspondence settings for estimating depth [46], optical flow [9, 51], or similarity transformations [59, 21], our objective is to infer a field of affine transformations, each represented by a 2×3 matrix

$$\mathbf{T}_i = \begin{bmatrix} \mathbf{T}_{i,\mathbf{x}} \\ \mathbf{T}_{i,\mathbf{y}} \end{bmatrix} \tag{1}$$

that maps pixel i to $i' = \mathbf{T}_i\mathbf{i}$, where $\mathbf{i}$ is pixel i represented in homogeneous coordinates such that $\mathbf{i} = [i, 1]^T$.

In this work, we solve for affine transformations that may lie anywhere in the continuous solution space. This is made possible by formulating the inference of dense affine transformation fields as a discrete optimization problem with continuous regularization. This optimization seeks to minimize an energy of the form

$$E(\mathbf{T}) = E_{data}(\mathbf{T}) + \lambda E_{smooth}(\mathbf{T}), \tag{2}$$

consisting of a data term that accounts for matching evidence between descriptors and a smoothness term that favors similar affine transformations among adjacent pixels with a balancing parameter λ.

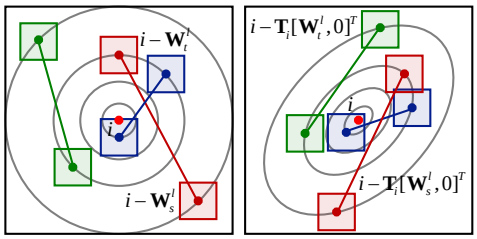
Our data term is defined as follows:

$$E_{data}(\mathbf{T}) = \sum_i \sum_{j \in \mathcal{N}_i} \omega_{ij}^I \min(\|\mathcal{D}_j - \mathcal{D}'_{j'}(\mathbf{T}_i)\|_1, \tau). \tag{3}$$

It is designed to estimate the affine transformation $\mathbf{T}_i$ by aggregating the matching costs of descriptors between neighboring pixels j and transformed pixels $j' = \mathbf{T}_i\mathbf{j}$ within a local aggregation window $\mathcal{N}_i$. A truncation threshold τ is used to deal with outliers and occlusions. It should be noted that aggregated data terms have been popularly used in stereo [46] and optical flow [27]. For dense semantic correspondence, several methods have employed aggregated data terms; however, they often produce undesirable results across object boundaries due to uniform weights that ignore image structure [23, 21], or fail to deal with geometric distortions like affine transformations as they rely on a regular grid structure for local aggregation windows [59]. By contrast, the proposed method adaptively aggregates matching costs using edge-preserving bilateral weights ω_{ij}^I as in [55, 19] on a geometrically-variant grid structure in order to produce spatially smooth yet discontinuity-preserving labeling results even under affine transformations.

Our smoothness term is defined as follows to regularize affine transformation fields $\mathbf{T}_i$ within a local neighborhood:

$$E_{smooth}(\mathbf{T}) = \sum_i \sum_{j \in \mathcal{M}_i} \upsilon_{ij}^I \|\mathbf{T}_i\mathbf{j} - \mathbf{T}_j\mathbf{j}\|^2. \tag{4}$$



(a) FCSS [24] (b) Affine-FCSS

Figure 2. Illustration of (a) FCSS descriptor [24] and (b) affine-FCSS descriptor. Within a support window, sampling patterns $\mathbf{W}_s^l$ and $\mathbf{W}_t^l$ are transformed according to affine fields $\mathbf{T}_i$.

When the affine transformation $\mathbf{T}$ is constrained to $[\mathbf{I}_{2\times2}, \mathbf{u}]$ with $\mathbf{u} = [u_{\mathbf{x}}, u_{\mathbf{y}}]^T$ and $\mathcal{M}_i$ is the 4-neighborhood, this smoothness term becomes the first order derivative of the optical flow vector as in many conventional methods [33, 38]. However, non-rigid deformations occur with high frequency in semantic correspondence, and such a basic constraint is inadequate for modeling the smoothness of affine transformation fields. Our smoothness term is formulated to address this by regularizing estimated affine transformations $\mathbf{T}_i$ in a moving least squares manner [45] within local neighborhood $\mathcal{M}_i$. We define the smoothness constraint of affine transformation fields by fitting $\mathbf{T}_i$ based on the affine flow fields of neighboring pixels $\mathbf{T}_j\mathbf{j}$. Unlike conventional moving least square solvers [45], our smoothness term incorporates edge-preserving bilateral weights υ_{ij}^I as in [55, 19] for image structure-aware regularization.

Minimizing the energy in (2) is a non-convex optimization problem defined over an infinite continuous solution space. A similar issue exists for optical flow estimation [8, 58, 42]. To minimize the non-convex energy function, several techniques such as a hybrid method with descriptor matching [8, 42] and a coarse-to-fine scheme [58] have been used, but they are tailored to optical flow estimation and have exhibited limited performance. We instead use a penalty decomposition scheme to alternately solve for the discrete and continuous affine transformation fields. An efficient filter-based discrete optimization technique is used to locally estimate discrete affine transformations in a manner similar to PMF [37]. The weakness of the implicit smoothing in the discrete local optimization is overcome by regularizing the affine transformation fields through global optimization in the continuous space. This alternating optimization is repeated until convergence. Furthermore, to acquire matching evidence for semantic correspondence under spatially-varying affine fields, we extend the FCSS descriptor [24] to model affine variations.

3.2. Affine-FCSS Descriptor

To estimate a matching cost, a dense descriptor $\mathcal{D}_i$ is extracted over the local support window of each image point I_i. For this we employ the state-of-the-art FCSS descriptor

[24] for dense semantic correspondence, which formulates LSS [47] within a fully convolutional network in a manner where the patch sampling patterns and self-similarity measure are both learned. Formally, FCSS can be described as a vector of feature values $\mathcal{D}_i = \bigcup_l \mathcal{D}_i^l$ for $l \in \{1, ..., L\}$ with the maximum number of sampling patterns L, where the feature values are computed as

$$\mathcal{D}_i^l = \exp(-\mathcal{S}(i - \mathbf{W}_s^l, i - \mathbf{W}_t^l)/\mathbf{W}_\sigma). \quad (5)$$

$\mathcal{S}(\cdot, \cdot)$ represents the self-similarity between two convolutional activations taken from a sampling pattern around center pixel i, and can be expressed as

$$\mathcal{S}(i - \mathbf{W}_s^l, i - \mathbf{W}_t^l) = \|\mathcal{F}(\mathbf{A}_i; \mathbf{W}_s^l) - \mathcal{F}(\mathbf{A}_i; \mathbf{W}_t^l)\|^2, \quad (6)$$

where $\mathcal{F}(\mathbf{A}_i; \mathbf{W}_s^l) = \mathbf{A}_{i-\mathbf{W}_s^l}$ and $\mathcal{F}(\mathbf{A}_i; \mathbf{W}_t^l) = \mathbf{A}_{i-\mathbf{W}_t^l}$, $\mathbf{W}_s^l = [W_{s,\mathbf{x}}^l, W_{s,\mathbf{y}}^l]$ and $\mathbf{W}_t^l = [W_{t,\mathbf{x}}^l, W_{t,\mathbf{y}}^l]$ compose the l-th learned sampling pattern, and $\mathbf{A}_i$ is the convolutional activation through feed-forward process $\mathcal{F}(I_i; \mathbf{W}_c)$ for I_i with network weights $\mathbf{W}_c$. The network parameters $\mathbf{W}_c$, $\mathbf{W}_s$, $\mathbf{W}_t$, and $\mathbf{W}_\sigma$ are learned in an end-to-end manner to provide optimal correspondence performance.

The FCSS descriptor provides high invariance to appearance variations, but it inherently cannot deal with geometric variations due to its pre-defined sampling patterns for all pixels in an image. Furthermore, although its computation is efficient, FCSS cannot in practice be evaluated exhaustively over all the affine candidates during optimization. To alleviate these limitations, we extend the FCSS descriptor to adapt to affine transformation fields. This is accomplished by reformulating the sampling patterns so that they account for the affine transformations. To expedite this computation, we first compute $\mathbf{A}_i$ over the entire image domain by passing it through the network. An FCSS descriptor $\mathcal{D}_i(\mathbf{T}_i)$ transformed under an affine field $\mathbf{T}_i$ can then be built by computing self-similarity on transformed sampling patterns

$$\|\mathcal{F}(\mathbf{A}_i; \mathbf{T}_i[\mathbf{W}_s^l, 0]^T) - \mathcal{F}(\mathbf{A}_i; \mathbf{T}_i[\mathbf{W}_t^l, 0]^T)\|^2. \quad (7)$$

With this approach, repeated computation of convolutional activations over different affine transformations of the image can be avoided. The affine transformation is efficiently inferred in a discrete optimization described in the following section. Differences between the FCSS descriptor and the affine-FCSS descriptor are illustrated in Fig. 2.

3.3. Solution

Since affine transformation fields are defined in an infinite label space, minimizing our energy function $E(\mathbf{T})$ directly is infeasible. Through fine-scale discretization of this space, affine transformation fields could be estimated through discrete global optimization, but at a tremendous computational cost. To address this issue, we introduce an auxiliary affine field $\mathbf{L}$ to decouple our data and regularization terms, and approximate the original minimization problem as the following auxiliary energy formulation:

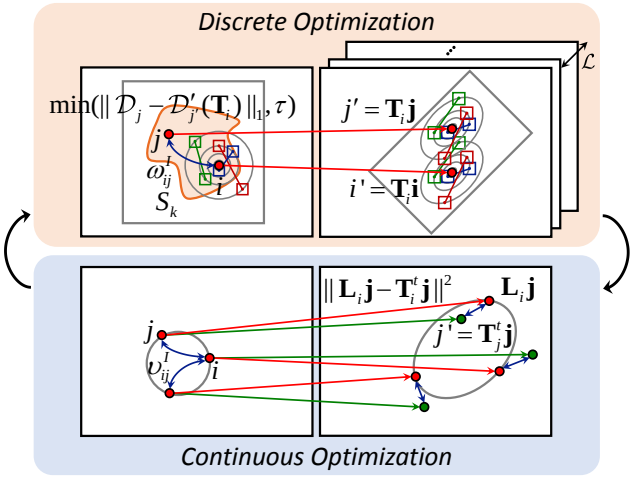


Figure 3. Our DCTM method consists of discrete optimization and continuous optimization. Our DCTM method differs from the conventional PMF [37] by alternately optimizing the discrete label space and performing the continuous regularization.

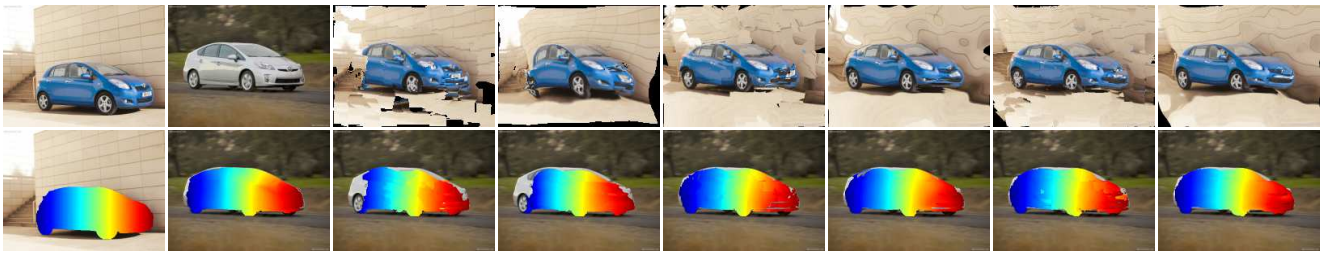
$$E_{\text{aux}}(\mathbf{T}, \mathbf{L}) = \sum_i \sum_{j \in \mathcal{N}_i} \omega_{ij}^I \min(\|\mathcal{D}_j - \mathcal{D}'_{j'}(\mathbf{T}_i)\|_1, \tau) + \mu \sum_i \|\mathbf{L}_i - \mathbf{T}_i\|^2 + \lambda \sum_i \sum_{j \in \mathcal{M}_i} \upsilon_{ij}^I \|\mathbf{L}_i\mathbf{j} - \mathbf{T}_j\mathbf{j}\|^2. \quad (8)$$

Since this energy function is based on two affine transformations, $\mathbf{L}$ and $\mathbf{T}$, we employ alternating minimization to solve for them and boost matching performance in a synergistic manner. We split the optimization of $E_{\text{aux}}(\mathbf{L}, \mathbf{T})$ into two sub-problems, namely a discrete local optimization problem with respect to $\mathbf{T}$ and a continuous global optimization problem with respect to $\mathbf{L}$. Increasing μ through the iterations drives the affine fields $\mathbf{T}$ and $\mathbf{L}$ together and eventually results in $\lim_{\mu \to \infty} E_{\text{aux}} \approx E$.

Discrete Optimization To infer the discrete affine transformation field $\mathbf{T}^t$ with $\mathbf{L}^{t-1}$ being fixed at the t-th iteration, we first discretize the continuous parameter space and then solve the problem through filter-based label inference. For discrete affine transformation candidates $\mathbf{T} \in \mathcal{L}$, the matching cost between FCSS descriptors $\mathcal{D}_j$ and $\mathcal{D}'_{j'}(\mathbf{T})$ is first measured as

$$C_j(\mathbf{T}) = \min(\|\mathcal{D}_j - \mathcal{D}'_{j'}(\mathbf{T})\|_1, \tau), \quad (9)$$

where $\mathcal{D}'_{j'}(\mathbf{T})$ is the affine-FCSS descriptor with respect to $\mathbf{T}$. This yields an affine-invariant matching cost. Furthermore, since j' varies according to affine fields such that $j' = \mathbf{T}\mathbf{j}$, affine-varying regular grids can be used when aggregating matching costs, thus enabling affine-invariant cost



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 4. DCTM convergence: (a) Source image; (b) Target image; Iterative evolution of warped images (c), (e), (g) after discrete optimization and (d), (f), (h) after continuous optimization. Our DCTM optimizes the label space with continuous regularization during the iterations, which facilitates convergence and boosts matching performance.

aggregation. To aggregate the raw matching costs, we apply EAF on $C_i(\mathbf{T})$ such that

$$\bar{C}_i(\mathbf{T}) = \sum\nolimits_{j \in \mathcal{N}_i} \omega_{ij}^I C_j(\mathbf{T}), \tag{10}$$

where ω_{ij}^I is the normalized adaptive weight of a support pixel j, which can be defined in various ways with respect to the structures of the image I [55, 14, 19].

In determining the affine field $\mathbf{T}$, the matching costs are also augmented by the previously estimated affine transformation field $\mathbf{L}_i^{t-1}$ such that

$$G_i(\mathbf{T}) = \mu \|\mathbf{T} - \mathbf{L}_i^{t-1}\|^2 + \lambda \sum_{j \in \mathcal{M}_i} \upsilon_{ij}^I \|\mathbf{T}\mathbf{j} - \mathbf{L}_i^{t-1}\mathbf{j}\|^2. \tag{11}$$

Since $\|\mathbf{T}\mathbf{j} - \mathbf{L}_i^{t-1}\mathbf{j}\|^2 = \|(\mathbf{T} - \mathbf{L}_i^{t-1})\mathbf{j}\|^2$ and $\mathbf{T} - \mathbf{L}_i^{t-1}$ is independent to pixel j within the support window, $G_i(\mathbf{T})$ can be efficiently computed by using constant-time EAF, as described in detail in the supplementary material.

The resultant label at the t-th iteration is determined with a winner-takes-all (WTA) scheme:

$$\mathbf{T}_i^t = \text{argmin}_{\mathbf{T} \in \mathcal{L}} \{\bar{C}_i(\mathbf{T}) + G_i(\mathbf{T})\}. \tag{12}$$

Continuous Optimization To solve the continuous affine transformation field $\mathbf{L}^t$ with $\mathbf{T}^t$ being fixed, we formulate the problem as an image warping minimization:

$$\sum_i \left(\mu \|\mathbf{L}_i - \mathbf{T}_i^t\|^2 + \lambda \sum_{j \in \mathcal{M}_i} \upsilon_{ij}^I \|\mathbf{L}_i\mathbf{j} - \mathbf{T}_j^t\mathbf{j}\|^2 \right). \tag{13}$$

Since this involves solving spatially-varying weighted least squares at each pixel i, the computational burden inevitably increases when considering non-local neighborhoods $\mathcal{M}_i$. To expedite this, existing MLS solvers adopted grid-based sampling [45] at the cost of quantization errors or parallel processing [22] with additional hardware. In contrast, our method optimizes the objective with a sparse matrix solver, yielding a substantial runtime gain. Since the $\mathbf{L}_i\mathbf{j}$ term can be formulated in the $\mathbf{x}$- and $\mathbf{y}$-directions separatively, $[\mathbf{L}_{i,\mathbf{x}}\mathbf{j}, \mathbf{L}_{i,\mathbf{y}}\mathbf{j}]^T$, we decompose the objective into

Algorithm 1: DCTM Framework

Input: images I, I', FCSS network parameter $\mathbf{W}$
Output: dense affine transformation fields $\mathbf{T}$
Parameters: number of segments K, pyramid levels F

/ **Initialization** */*
Partition I into a set of disjoint K segments $\{S_k\}$
Initialize affine fields as $\mathbf{T}_i = [\mathbf{I}_{2\times2}, \mathbf{0}_{2\times1}]$
for $f = 1 : F$ **do**
  Build convolution activations $\mathbf{A}^f$, $\mathbf{A}'^f$ for I^f, I'^f
  Initialize affine fields $\mathbf{T}_i^f = \mathbf{L}_i^{f-1}$ when $f > 2$
  while not converged **do**
    / **Discrete Optimization** */*
    Initialize affine fields $\mathbf{T}_i^t = \mathbf{L}_i^{t-1}$
    for $k = 1 : K$ **do**
      / **Propagation** */*
      For S_k, construct affine candidates $\mathbf{T} \in \mathcal{L}_p$ from neighboring segments
      Build cost volumes $\bar{C}_i(\mathbf{T})$ and $G_i(\mathbf{T})$
      Determine $\mathbf{T}_i^t$ using (12)
      / **Random Search** */*
      Construct affine candidates $\mathbf{T} \in \mathcal{L}_r$ from randomly sampled affine fields
      Determine $\mathbf{T}_i^t$ by Step **7**-**8**
    end for
    / **Continuous Optimization** */*
    Estimate affine fields $\mathbf{L}_i^t$ from $\mathbf{T}_i^t$ using (15)
  end while
end for

two separable energy functions. For the $\mathbf{x}$-direction, the energy function can be represented as

$$\sum_i \left(\mu \|\mathbf{L}_{i,\mathbf{x}} - \mathbf{T}_{i,\mathbf{x}}^t\|^2 + \lambda \sum_{j \in \mathcal{M}_i} \upsilon_{ij}^I \|\mathbf{L}_{i,\mathbf{x}}\mathbf{j} - \mathbf{T}_{j,\mathbf{x}}^t\mathbf{j}\|^2 \right). \tag{14}$$

By setting the gradient of this objective with respect to $\mathbf{L}_{\mathbf{x},i}$ to zero, the minimizer $\mathbf{L}_{i,\mathbf{x}}^t$ is obtained by solving a linear system based on a large sparse matrix:

$$(\mu/\lambda\mathbf{I} + \mathbf{U})\mathbf{L}_{\mathbf{x}}^t = (\mu/\lambda\mathbf{I} + \mathbf{K})\mathbf{T}_{\mathbf{x}}^t, \tag{15}$$

where $\mathbf{I}$ denotes a $3N \times 3N$ identity matrix with N denoting the number of pixels in image I. $\mathbf{L}^t_{\mathbf{x}}$ and $\mathbf{T}^t_{\mathbf{x}}$ denote $3N \times 1$ column vectors containing $\mathbf{L}^t_{i,\mathbf{x}}$ and $\mathbf{T}^t_{i,\mathbf{x}}$, respectively. $\mathbf{U}$ and $\mathbf{K}$ denote matrices defined as

$$\mathbf{U} = \begin{bmatrix} \psi(\mathbf{V}X^2) & \psi(\mathbf{V}XY) & \psi(\mathbf{V}X) \\ \psi(\mathbf{V}XY) & \psi(\mathbf{V}Y^2) & \psi(\mathbf{V}Y) \\ \psi(\mathbf{V}X) & \psi(\mathbf{V}Y) & \mathbf{I}_{N\times N} \end{bmatrix}, \quad (16)$$

and

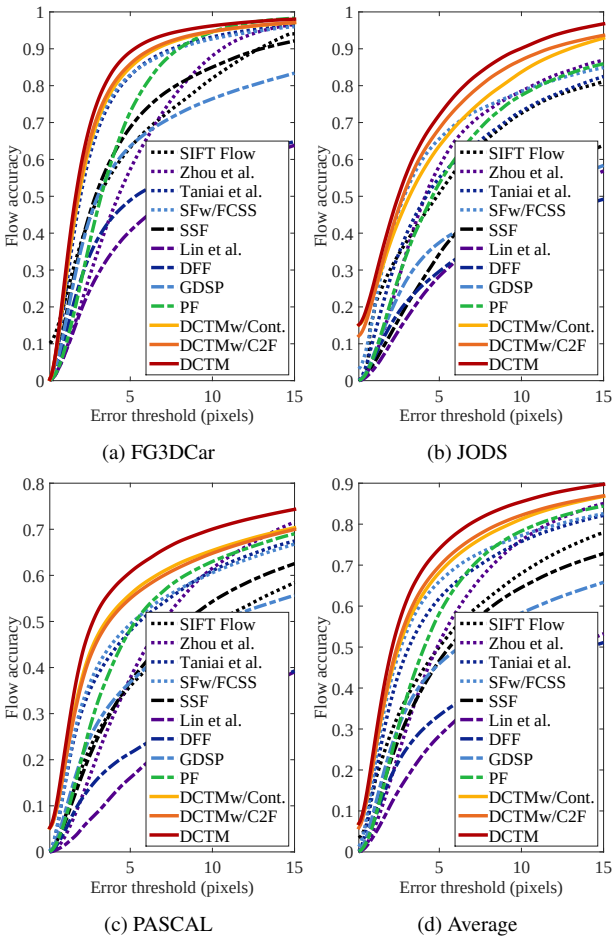
$$\mathbf{K} = \begin{bmatrix} \mathbf{V}\psi(X) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}\psi(Y) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{V} \end{bmatrix}, \quad (17)$$

where $\mathbf{V}$ is an $N \times N$ kernel matrix whose nonzero elements are given by the weights υ^I_{ij}, $\psi(\cdot)$ denotes a diagonalizaition operator, X and Y denote $N \times 1$ column vectors containing $i_{\mathbf{x}}$ and $i_{\mathbf{y}}$, respectively. $X^2 = X \circ X$, $Y^2 = Y \circ Y$, and $XY = X \circ Y$, where $\circ$ denotes the Hadamard product.

Since υ^I_{ij} is a normalized bilateral weight, the matrices $\mathbf{U}$ and $\mathbf{K}$ can be efficiently computed using recent EAF algorithms [14, 19]. Furthermore, since $\mu/\lambda\mathbf{I} + \mathbf{U}$ is a block-diagonal matrix, $\mathbf{L}^t_{\mathbf{x}}$ can be estimated efficiently using a fast sparse matrix solver [25]. After optimizing $\mathbf{L}^t_{\mathbf{y}}$ in a similar manner, we then have the continuous affine fields $\mathbf{L}^t$.

Iterative Inference In our filter-based discrete optimization, exhaustively evaluating the raw and aggregated costs for every label $\mathcal{L}$ is still prohibitively time-consuming. Thus we utilize the PMF [37] which jointly leverages label cost filtering and fast randomized PatchMatch search in a high dimensional label space. Our discrete optimization differs from the PMF by optimizing the discrete label space with continuous regularization during the iterations, which facilitates convergence and boosts matching performance.

We first decompose an image I into a set of K disjoint segments $I = \{S_k, k = 1, ..., K\}$ and build its set of spatially adjacent segment neighbors. Then for each segment S_k, two sets of label candidates from the *propagation* and *random search* steps are evaluated for each graph node in scan order. In the propagation step, for each segment S_k, a candidate pixel i is randomly sampled from each neighboring segment, and a set of current best labels $\mathcal{L}_p$ for i is defined by $\{\mathbf{T}_i\}$. For these $\mathcal{L}_p$, EAF-based cost aggregation is then performed for the segment S_k. In the random search step, a center-biased random search as done in PatchMatch [2] is performed for the current segment S_k, where a sequence of random labels $\mathcal{L}_r$ sampled around the current best label is evaluated. After an iteration of the propagation and random search steps for all segments, we apply continuous optimization as described in the preceding section to regularize the discrete affine transformation fields. After each iteration, we enlarge μ such that $\mu \leftarrow c\mu$ with a constant value $1 < c \leq 2$ to accelerate convergence. Fig. 3 summarizes our DCTM method, consisting of discrete and continuous optimization, and Fig. 4 illustrates the convergence of our DCTM method.



(a) FG3DCar (b) JODS

(c) PASCAL (d) Average

Figure 6. Average flow accuracy with respect to endpoint error threshold on the Taniai benchmark [53].

To boost matching performance and convergence of our algorithm, we apply our method in a coarse-to-fine manner, where images I^f are constructed at F image pyramid levels $f = \{1, ..., F\}$ and affine transform fields $\mathbf{T}^f$ are predicted at level f. Coarser scale results are then used as initialization for the finer levels. Algorithm 1 provides a summary of the overall procedure of our DCTM method.

4. Experimental Results

4.1. Experimental Settings

For our experiments, we used the FCSS descriptor provided by authors, which is learned on Caltech-101 dataset [12]. For EAF for ω^I_{ij} and υ^I_{ij}, we utilized the guided filter [20], where the radius and smoothness parameters are set to $\{16, 0.01\}$. The weights in energy function were initially set to $\{\lambda, \mu\} = \{0.01, 0.1\}$ by cross-validation, but μ increases as evolving iterations with $c = 1.8$. The SLIC [1] segment number K increases sublinearly with the im-

(a) (b) (c) (d) (e) (f) (g) (h)

Figure 5. Qualitative results on the Taniai benchmark [53]: (a) source image, (b) target image, (c) Lin *et al.* [29], (d) DFF [59], (e) PF [16], (f) Taniai *et al.* [53], (g) SF w/FCSS [24], and (h) DCTM. The source images were warped to the target images using correspondences.

Methods	FG3D	JODS	PASC.	Avg.
SIFT Flow [33]	0.632	0.509	0.360	0.500
DSP [23]	0.487	0.465	0.382	0.445
Zhou *et al.* [61]	0.721	0.514	0.436	0.556
Taniai *et al.* [53]	0.830	0.595	0.483	0.636
SF w/DAISY [54]	0.636	0.373	0.338	0.449
SF w/VGG [49]	0.756	0.490	0.360	0.535
SF w/FCSS [24]	0.830	0.653	0.494	0.660
SLS [18]	0.525	0.519	0.320	0.457
SSF [41]	0.687	0.344	0.370	0.467
SegSIFT [56]	0.612	0.421	0.331	0.457
Lin *et al.* [29]	0.406	0.283	0.161	0.283
DFF [59]	0.489	0.296	0.214	0.333
GDSP [21]	0.639	0.374	0.368	0.459
Proposal Flow [16]	0.786	0.653	0.531	0.657
DCTM w/DAISY	0.710	0.506	0.482	0.566
DCTM w/VGG	0.790	0.611	0.528	0.630
DCTM wo/Cont.	0.850	0.637	0.559	0.682
DCTM wo/C2F	0.859	0.684	0.550	0.698
DCTM	**0.891**	**0.721**	**0.610**	**0.740**

Table 1. Matching accuracy compared to state-of-the-art correspondence techniques on the Taniai benchmark [53].

age size, *e.g.*, $K = 500$ for 640×480 images. The image pyramid level F is set to 3. We implemented our DCTM method in Matlab/C++ on Intel Core i7-3770 CPU at 3.40 GHz, and measured the runtime on a single CPU core. Our code will be made publicly available.

In the following, we comprehensively evaluated our DCTM method through comparisons to the state-of-the-art methods for dense semantic correspondences, including SIFT Flow [33], DSP [23], Zhou *et al.* [61], UCN [11], Taniai *et al.* [53], SIFT Flow optimization with VGG[1] [49] and FCSS [24] descriptor. Furthermore geometric-invariant methods including SLS [18], SSF [41], SegSIFT [56], Lin *et al.* [29], DFF [59], GDSP [21], and PF [16] were evaluated. The performance was measured on Taniai benchmark [53], Proposal Flow dataset [16], and PASCAL-VOC dataset [10]. To validate the components of our method, we additionally examined the performance contributions of the continuous optimization (wo/Cont.) and the coarse-to-fine scheme (wo/C2F). Furthermore the performance of our DCTM method when combined with other dense descriptors[2] was examined using the DAISY [54] and VGG [49].

[1]In the 'VGG', ImageNet pretrained VGG-Net [49] from the botton conv1 to the conv3-4 layer were used with L_2 normalization [50].

4.2. Results

Taniai Benchmark [53] We first evaluated our DCTM method on the Taniai benchmark [53], which consists of 400 image pairs divided into three groups: FG3DCar [31], JODS [44], and PASCAL [17]. As in [53, 24], flow accuracy was measured by computing the proportion of foreground pixels with an absolute flow endpoint error that is smaller than a certain threshold T, after resizing images so that its larger dimension is 100 pixels.

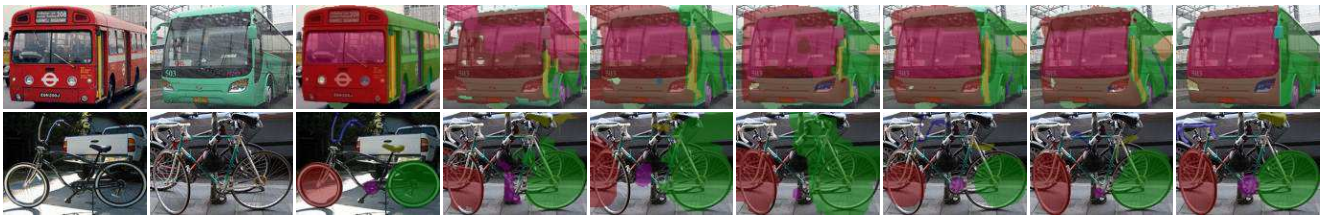
Table 1 summarizes the matching accuracy for state-of-the-art correspondence techniques ($T = 5$ pixels). Fig. 5 displays qualitative results for dense flow estimation. Fig. 6 plots the flow accuracy with respect to error threshold. Compared to methods based on handcrafted features [41, 59, 21], CNN based methods [53, 24] provide higher accuracy even though they do not consider geometric variations. The method of Lin *et al.* [29] cannot estimate reliable correspondences due to unstable sparse correspondences. Thanks to its discrete labeling optimization with continuous regularization and affine-FCSS, our DCTM method provides state-of-the-art performance.

Proposal Flow Benchmark [16] We also evaluated our FCSS descriptor on the Proposal Flow benchmark [16], which includes 10 object sub-classes with 10 keypoint annotations for each image. For the evaluation metric, we used the probability of correct keypoint (PCK) between flow-warped keypoints and the ground truth [34, 16]. The warped keypoints are deemed to be correctly predicted if they lie within $\alpha \cdot \max(H, W)$ pixels of the ground-truth keypoints for $\alpha \in [0, 1]$, where H and W are the height and width of the object bounding box, respectively. The PCK values

[2]These experiments use only the upright version of the descriptors.

(a) (b) (c) (d) (e) (f) (g) (h)

Figure 7. Qualitative results on the Proposal Flow benchmark [16]: (a) source image, (b) target image, (c) SSF [41], (d) DSP [23], (e) GDSP [21], (f) PF [16], (g) SF w/FCSS [24], and (h) DCTM. The source images were warped to the target images using correspondences.



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 8. Visualizations of dense flow field with color-coded part segments on the PASCAL-VOC part dataset [10]: (a) source image, (b) target image, (c) source mask, (d) DFF [59], (e) GDSP [21], (f) Zhou *et al.* [61], (g) SF w/FCSS [24], (h) DCTM, and (i) target mask.

Methods	PCK		
	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.15$
SIFT Flow [33]	0.247	0.380	0.504
DSP [23]	0.239	0.364	0.493
Zhou *et al.* [61]	0.197	0.524	0.664
SF w/FCSS [24]	0.354	0.532	0.681
SSF [41]	0.292	0.401	0.531
Lin *et al.* [29]	0.192	0.354	0.487
DFF [59]	0.241	0.362	0.510
GDSP [21]	0.242	0.487	0.512
Proposal Flow [16]	0.284	0.568	0.682
DCTM	**0.381**	**0.610**	**0.721**

Table 2. Matching accuracy compared to state-of-the-art correspondence techniques on the Proposal Flow benchmark [16].

were measured for different correspondence techniques in Table 2. Fig. 7 shows qualitative results for dense flow estimation. Our DCTM method exhibits performance competitive to the state-of-the-art correspondence techniques.

PASCAL-VOC Parts Dataset [10] Lastly, we evaluated our DCTM method on the dataset provided by [62], where the images are sampled from the PASCAL parts dataset [10]. With human-annotated part segments, we measured part matching accuracy using the weighted intersection over union (IoU) score between transferred segments and ground truths, with weights determined by the pixel area of each part. To evaluate alignment accuracy, we measured the PCK metric using keypoint annotations for the 12 rigid PASCAL classes [57]. Table 3 summarizes the matching accuracy compared to state-of-the-art correspondence methods. Fig. 8 visualizes estimated dense flow with color-coded part segments. From the results, our DCTM method is found to yield the highest matching accuracy.

Methods	IoU	PCK	
		$\alpha = 0.05$	$\alpha = 0.1$
Zhou *et al.* [61]	-	-	0.24
UCN [11]	-	0.26	0.44
SF w/ FCSS [33]	0.44	0.28	0.47
DFF [59]	0.36	0.14	0.31
GDSP [21]	0.40	0.16	0.34
Proposal Flow [16]	0.41	0.17	0.36
DCTM	**0.48**	**0.32**	**0.50**

Table 3. Matching accuracy on the PASCAL-VOC dataset [10].

Computation Speed For all the test cases, our DCTM method converges with 3-5 iterations on each image pyramid level. For 320×240 images, the average runtime of DCTM is 15-20 seconds, compared to 216 seconds for GDSP [21], 73 seconds for DFF [59], 276 seconds for Lin *et al.* [29], and 321 seconds for Taniai *et al.* [53].

5. Conclusion

We presented DCTM, which estimates dense affine transformation fields through a discrete label optimization in which the labels are iteratively updated via continuous regularization. DCTM infers solutions from the continuous space of affine transformations in a manner that can be computed efficiently through constant-time edge-aware filtering and the affine-FCSS descriptor. A direction for further study is to examine how the semantic flow of DCTM could benefit single-image 3D reconstruction and instance-level object segmentation.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *TPAMI*, 34(11):2274–2282, 2012.

[2] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein. The generalized patchmatch correspondence algorithm. *In: ECCV*, 2010.

[3] F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz. Pmbp: Patchmatch belief propagation for correspondence field estimation. *IJCV*, 110(1):2–13, 2014.

[4] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of image defromations. *TPAMI*, 11(6):567–585, 1989.

[5] N. K. Bose and N. A. Ahuja. Super-resolution and noise filtering using moving least squares. *TIP*, 15(8):2239–2248, 2006.

[6] Y. Boykov, O. Yeksler, and R. Zabih. Fast approximation enermgy minimization via graph cuts. *TPAMI*, 23(11):1222–1239, 2001.

[7] H. Bristow, J. Valmadre, and S. Lucey. Dense semantic correspondence where every pixel is a classifier. *In: ICCV*, 2015.

[8] T. Brox, C. Bregler, and J. Malik. Large displacement optical flow. *In: CVPR*, 2009.

[9] D. Butler, J. Wulff, G. Stanley, and M. Black. A naturalistic open source movie for optical flow evaluation. *In: ECCV*, 2012.

[10] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasum, and A. Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. *In: CVPR*, 2014.

[11] C. B. Choy, Y. Gwak, and S. Savarese. Universal correspondence network. *In:NIPS*, 2016.

[12] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *TPAMI*, 28(4):594–611, 2006.

[13] S. Fleishman, D. Cohen-Or, and C. T. Silva. Robust moving least squares fitting with sharp features. *In: SIGGRAPH*, 2005.

[14] E. Gastal and M. Oliveira. Domain transform for edge-aware image and video processing. *In: SIGGRAPH*, 2011.

[15] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski. Non-rigid dense correspondence with applications for image enhancement. *In: SIGGRAPH*, 2011.

[16] B. Ham, M. Cho, C. Schmid, and J. Ponce. Proposal flow. *In: CVPR*, 2016.

[17] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. *In: ICCV*, 2011.

[18] T. Hassner, V. Mayzels, and L. Zelnik-Manor. On sifts and their scales. *In: CVPR*, 2012.

[19] K. He, J. Sun, and X. Tang. Guided image filtering. *In: ECCV*, 2010.

[20] K. He, J. Sun, and X. Tang. Guided image filtering. *TPAMI*, 35(6):1397–1409, 2013.

[21] J. Hur, H. Lim, C. Park, and S. C. Ahn. Generalized deformable spatial pyramid: Geometry-preserving dense correspondence estimation. *In: CVPR*, 2015.

[22] Y. Hwang, J. Lee, I. Kweon, and S. Kim. Color transfer using probabilistic moving least squares. *In: CVPR*, 2014.

[23] J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspondences. *In: CVPR*, 2013.

[24] S. Kim, D. Min, B. Ham, S. Jeon, S. Lin, and K. Sohn. Fcss: Fully convolutional self-similarity for dense semantic correspondence. *In: CVPR*, 2017.

[25] D. Krishnan, R. Fattal, and R. Szeliski. Efficient preconditioning of laplacian matrices for computer graphics. *In: SIGGRAPH*, 2013.

[26] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Math. Comp.*, 87:141–158, 1981.

[27] Y. Li, D. Min, M. S. Brown, M. N. Do, and J. Lu. Spm-bp: Sped-up patchmatch belief propagation for continuous mrfs. *In: ICCV*, 2015.

[28] W. Y. Lin, M. M. Cheng, S. Zheng, J. Lu, and N. Crook. Pm-huber: Patchmatch with huber regularization for stereo matching. *In: ICCV*, 2013.

[29] W. Y. Lin, L. Liu, Y. Matsushita, K. L. Low, and S. Liu. Aligning images in the wild. *In: CVPR*, 2012.

[30] W. Y. Lin, S. Liu, Y. Matsushita, T. T. Ng, and L. F. Cheong. Smoothly varying affine stitching. *In: CVPR*, 2011.

[31] Y. L. Lin, V. I. Morariu, W. Hsu, and L. S. Davis. Jointly optimizing 3d model fitting and fine-grained classification. *In: ECCV*, 2014.

[32] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *TPAMI*, 33(12):2368–2382, 2011.

[33] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Trans. PAMI*, 33(5):815–830, 2011.

[34] J. Long, N. Zhang, and T. Darrell. Do convnets learn correspondence? *In: NIPS*, 2014.

[35] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[36] J. Lu, K. Shi, D. Shi, L. Lin, and M. N. Do. Cross-based local multipoint filtering. *In: CVPR*, 2012.

[37] J. Lu, H. Yang, D. Min, and M. N. Do. Patchmatch filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation. *In: CVPR*, 2013.

[38] D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, and M. N. Do. Fast global image smoothing based on weighted least squares. *TIP*, 23(12):5638–5653, 2014.

[39] A. Myronenko, X. Song, and M. Carreira-Perpinan. Non-rigid point set registration: Coherent point drift. *In: NIPS*, 2007.

[40] Online. `http://www.shapenet.org/`.

[41] W. Qiu, X. Wang, X. Bai, A. Yuille, and Z. Tu. Scale-space sift flow. *In: WACV*, 2014.

[42] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. *In: CVPR*, 2015.

[43] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *In: CVPR*, 2011.

[44] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu. Unsupervised joint object discovery and segmentation in internet images. *In: CVPR*, 2013.

[45] S. Schaefer, T. McPhail, and J. Warren. Image deformation using moving least squares. *ACM TOG*, 25(3):533–540, 2006.

[46] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1):7–42, 2002.

[47] E. Schechtman and M. Irani. Matching local self-similarities across images and videos. *In: CVPR*, 2007.

[48] A. Shekhovtsov, I. Kovtun, and V. Hlavac. Efficient mrf deformation model for non-rigid image matching. *In: CVPR*, 2007.

[49] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *In: ICLR*, 2015.

[50] H. O. Song, Y. Xiang, S. Jegelk, and S. Savarese. Deep metric learing via lifted structured feature embedding. *In: CVPR*, 2016.

[51] D. Sun, S. Roth, and M. J. Black. Secret of optical flow estimation and their principles. *In: CVPR*, 2010.

[52] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Trans. PAMI*, 30(6):1068–1080, 2008.

[53] T. Taniai, S. N. Sinha, and Y. Sato. Joint recovery of dense correspondence and cosegmentation in two images. *In: CVPR*, 2016.

[54] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *TPAMI*, 32(5):815–830, 2010.

[55] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. *In: ICCV*, 1998.

[56] E. Trulls, I. Kokkinos, A. Sanfeliu, and F. M. Noguer. Dense segmentation-aware descriptors. *In: CVPR*, 2013.

[57] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. *In: WACV*, 2014.

[58] L. Xu, J. J., and Y. Matsushita. Motion detial preserving optical flow estimation. *In: CVPR*, 2010.

[59] H. Yang, W. Y. Lin, and J. Lu. Daisy filter flow: A generalized discrete approach to dense correspondences. *In: CVPR*, 2014.

[60] A. L. Yuille and N. M. Grywacz. The motion coherence theory. *In: ICCV*, 1988.

[61] T. Zhou, P. Krahenbuhl, M. Aubry, Q. Huang, and A. A. Efros. Learning dense correspondence via 3d-guided cycle consistency. *In: CVPR*, 2016.

[62] T. Zhou, Y. J. Lee, S. X. Yu, and A. A. Efros. Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. *In: CVPR*, 2015.