Discrete-Continuous Transformation Matching for Dense Semantic Correspondence

Seungryong Kim, *Member, IEEE,* Dongbo Min, *Senior Member, IEEE,* Stephen Lin, *Member, IEEE,* and Kwanghoon Sohn*, *Senior Member, IEEE*

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. Furthermore, leveraging correspondence consistency and confidence-guided filtering in each iteration facilitates the convergence of our method. Experimental results show that this model outperforms the state-of-the-art methods for dense semantic correspondence on various benchmarks and applications.

Index Terms—Dense semantic correspondence, discrete optimization, continuous optimization, interative inference

1 Introduction

Establishing dense correspondences across *semantically* similar images is essential for numerous computer vision and computational photography applications, such as nonparametric scene parsing, scene recognition, image registration, semantic segmentation, or image editing [1], [2], [3]. Unlike traditional dense correspondence for estimating depth [4] or optical flow [5], [6], semantic correspondence poses additional challenges due to intra-class appearance and shape variations among different instances within the same object or scene category, which can degrade matching accuracy by conventional approaches [2], [7].

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 [8], [9], [10], [11]. However, geometric variations are considered in just a limited manner through conventional constraint settings such as those used for stereo matching or optical flow (i.e., translational motion only). Some methods have been proposed to solve more complex geometric variations such as scale or rotation [12], [13], [14], but they consider only a set of discretized scales and/or rotations as possible solutions, and do not capture the non-rigid geometric deformations that commonly exist between semantically similar images.

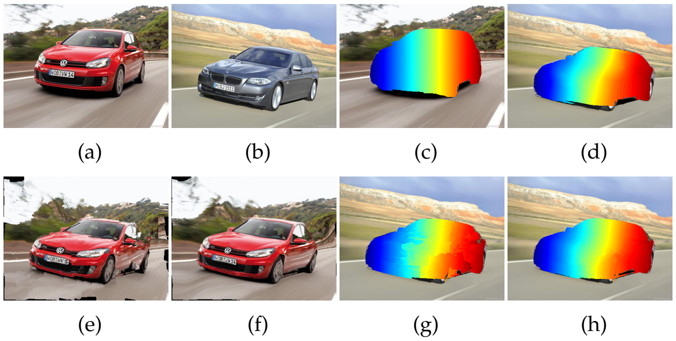


Fig. 1. Visualization of our DCTM results: (a) source image, (b) target image, (c), (d) ground truth correspondences, (e), (f), (g), (h) warped images and correspondences after discrete and continuous optimization, respectively. For semantically similar images undergoing non-rigid deformations, our DCTM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

It has been shown that these non-rigid image deformations can be locally well approximated by affine transformations [15], [16], [17]. 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 [18] or belief propagation (BP) [19], [20] and discrete local optimization such as constant-time filter-based cost volume filtering (CVF) [21] computationally infeasible. Although some methods such as PatchMatch belief propagation (PMBP) [22] and Sped-up PMBP (SPM-BP) [23] have been proposed for more efficient computation over large label spaces in discrete global optimization, they still cannot deal with extremely large search spaces such as that of affine transformations. In discrete local optimization, the PatchMatch Filter (PMF) [24] integrates constant-time

- *S. Kim and K. Sohn are with the School of Electrical and Electronic Engineering, Yonsei University, Seoul 120-749, Korea. E-mail: {srkim89, khsohn}@yonsei.ac.kr*
- *D. Min is with the Department of Computer Science and Engineering, Ewha Womans University, Seoul 03760, South Korea. E-mail: dbmin@ewha.ac.kr*
- *S. Lin is with Microsoft Research, Beijing 100080, China. E-mail: stevelin@microsoft.com*

* *Corresponding author*

filter-based CVF [21], [25] with a superpixel-based randomized search strategy inspired by PatchMatch [26] to reduce computational complexity with respect to the search space. PMF [24] is also leveraged for dense semantic correspondence in DAISY Filter Flow (DFF) [7], which finds labels for displacement fields as well as for scale and rotation. Extending DFF [7] to solve for affine transformations would be challenging though. One reason is that its efficient technique for computing DAISY features [27] at pre-determined scales and rotations cannot be directly applied for affine transformations defined over an infinite continuous solution space. Another reason is that, as shown in [14], [23], the weak implicit smoothness constraint embedded in PMF [24] makes it more susceptible to erroneous local minima, and this problem may be magnified in the higher-dimensional search space for affine transformations. An explicit smoothness constraint has been adopted to alleviate this problem in the context of stereo matching [28], but was 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 cast the inference of dense affine transformation fields as 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 optimization inspired by PMF [24], and then the discrete affine transformation field is globally regularized in a moving least squares manner [15]. 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, instead of simply selecting from a pre-defined, finite set of discrete label samples. We show that this continuous regularization additionally overcomes the aforementioned implicit smoothness constraint problem in PMF [24]. To further boost matching performance and convergence of our method, we also leverage correspondence consistency and an evolving cost aggregation based on confidence of correspondence in each iteration. Moreover, we model affine transformation fields directly within the fully convolutional self-similarity (FCSS) descriptor [11] in a manner that sampling patterns are reformulated to adapt to affine transformation fields. By efficiently reducing the repeated computations over computing FCSS descriptors on various affine transformations of the image, our approach achieves significant improvements in processing speed. Experimental results show that the presented model outperforms the latest methods for dense semantic correspondence on several benchmarks, including that of TSS [29], PF-WILLOW [30], PF-PASCAL [31], the CUB-200-2011 dataset [32], the PASCAL-VOC part dataset [33], and Caltech-101 [34].

This manuscript extends the conference version of this work [35]. It newly adds (1) an extension of DCTM based on correspondence consistency and confidence-guided filtering; (2) an in-depth analysis of DCTM; and (3) an extensive comparative study with existing semantic correspondence methods using various datasets. The source code of this work is available online at our project webpage: `http://diml.yonsei.ac.kr/∼srkim/DCTM/`.

2 Related Work

2.1 Semantic Correspondence

Most conventional techniques for dense semantic correspondence employ handcrafted features such as SIFT [36] and DAISY [27]. To elevate matching quality, they focus on improving optimization. Liu et al. pioneered the idea of dense correspondence across different scenes, and proposed SIFT flow (SF) [2], which is based on hierarchical dual-layer belief propagation. Inspired by this, Kim et al. [37] proposed the deformable spatial pyramid (DSP) which performs multi-scale regularization with a hierarchical graph. Hassner et al. [38] proposed a method to estimate correspondences between query and reference face images by regularizing the correspondence fields to produce similar semantic contents. More recently, Yang et al. [39] proposed the object-aware hierarchical graph (OHG) to regulate matching consistency over whole objects. Among other methods are those that take an exemplar-LDA approach [40], employ joint image set alignment [41], or jointly solve for cosegmentation [29]. Recently, Ham et al. [30], [31] presented the proposal flow (PF) algorithm to estimate semantic correspondences using object proposals. As all of these techniques use handcrafted descriptors, they lack the robustness to deformations that is possible with deep CNNs.

Recently, deep CNN-based descriptors have been used to establish semantic correspondences because of their high invariance to appearance variations. Pre-trained CNN features have been employed with the SIFT Flow [8]. Zhou et al. [10] proposed a deep network that exploits cycle-consistency with a 3-D CAD model [42] as a supervisory signal. Choy et al. [9] proposed the universal correspondence network (UCN) based on fully convolutional feature learning. Most recently, Novotny et al. [43] proposed AnchorNet that learns geometry-sensitive features for semantic correspondence with weak image-level labels. Kim et al. [11], [44] proposed the FCSS descriptor that formulates local self-similarity (LSS) [45] within a fully convolutional network. Because of its LSS-based structure, FCSS is inherently insensitive to intra-class appearance variations while maintaining precise spatial localization ability. Inspired by PF [30], Ufer et al. [46] proposed a method based on convolutional feature pyramids and activation-guided feature selection. Han et al. [47] proposed SCNet for learning a geometrically plausible model for semantic correspondence. Gaur et al. [48] proposed a novel optimization to re-purpose deep convolutional features to group semantically similar object parts. While these aforementioned techniques provide some amount of geometric invariance, none of them can deal with affine transformations across images, which frequently occur in dense semantic correspondence.

2.2 Transformation Invariance

Several methods aim to alleviate geometric variation problems in dense semantic correspondence through extensions of SF [2], including scale-less SF (SLS) [12], scale-space SF (SSF) [13], and generalized DSP (GDSP) [14]. However, these techniques have a critical practical limitation that their computational cost increases linearly with the search space size. Tau et al. [49] proposed a dense correspondence algorithm that propagates scales estimated from sparse interest points and uses them to optimize correspondence fields. However, since erroneous scales can be propagated from initial estimates, it has shown limited performance. A generalized PatchMatch algorithm [26] was proposed for efficient matching that leverages a randomized search scheme to avoid an exhaustive search for all possible solution spaces. It was utilized by HaCohen et al. [1] 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 [7], but its implicit smoothness constraint which relies on randomized sampling and propagation of good estimates in the direct neighborhood often induces mismatches. Recently, Rocco et al. [50], [51] proposed a CNN architecture for estimating a geometric matching (GMat) model that includes affine transformations. However, it only estimates globally-varying geometric fields, and thus exhibits limited performance in dealing with locally-varying geometric deformations. Moreover, deep CNN-based methods require a substantial learning procedure on large-scale training samples, limiting their applicability.

2.3 Image Manipulation

A possible approach for estimating dense correspondences is to interpolate sparsely matched points using thin plate splines (TPS) [52], motion coherence [16], [17], [53], or coherence point drift [54]. Moving least squares (MLS) is also a scattered point interpolation technique, first introduced in [55] to reconstruct a continuous function from a set of point samples by minimizing spatially-weighted least squares. MLS has been successfully used in applications such as image deformation [15], surface reconstruction [56], image super-resolution and denoising [57], or color transfer [58]. 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 are the methods of Lin et al. [16], [17], which estimate dense affine transformation fields constrained by global smoothness. However, they are formulated with sparse correspondences and also require considerable computation by applying 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 *semantically* similar images I and I', the objective of dense correspondence estimation is to establish a correspondence i' for each pixel $i = [i_{\mathbf{x}}, i_{\mathbf{y}}]$ in I. Unlike conventional dense correspondence settings for estimating depth [4], optical flow [5], [6], or similarity transformations (i.e., displacement, rotation, and uniform scale transformations) [7], [14], 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 dense affine transformation fields that may lie anywhere in the continuous solution space by minimizing 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 feature descriptors and a smoothness term that favors similar affine transformations among adjacent pixels with a balancing parameter λ.

3.1.1 Data Term

Our data term is defined as follows:

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

It represents matching evidence given an affine transformation $\mathbf{T}_i$ for each pixel i, by aggregating the matching costs between descriptors $\mathcal{D}_j$ and $\mathcal{D}'_{j'}(\mathbf{T}_i)$ of neighboring pixels j and transformed pixels $j' = \mathbf{T}_i \mathbf{j}$ within a local aggregation window $\mathcal{N}_i$ in a structure-aware manner. 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 matching [4], [28] and optical flow [23]. For dense semantic correspondence, some methods have also employed aggregated data terms; however, they often produce undesirable results across object boundaries due to uniform weights that ignore image structure [14], [37], or fail to deal with more complex geometric distortions like affine transformations as they rely on a square grid structure for local aggregation windows [7]. By contrast, the proposed method adaptively aggregates matching costs on a geometrically-variant grid structure using an adaptive weight ω^I_{ij} guided by the image I, e.g., $\omega^I_{ij} \propto \exp(-\|i-j\|^2/\sigma_r - \|I_i - I_j\|^2/\sigma_c)$, which measures geometric closeness and intensity similarity with parameters σ_r and σ_c as in [24], [59], [60]. It thus enables producing spatially smooth yet image discontinuity-preserving labeling results even under complex geometric deformations such as affine transformations.

3.1.2 Smoothness Term

Our smoothness term is defined to regularize affine transformation fields within a local neighborhood as follows:

$$E_{smooth}(\mathbf{T}) = \sum_i \sum_{u \in \mathcal{M}_i} \upsilon^I_{iu} \|\mathbf{T}_i \mathbf{u} - \mathbf{T}_u \mathbf{u}\|^2. \tag{4}$$

When the affine transformation $\mathbf{T}$ is constrained to $[\mathbf{I}_{2\times 2}, \mathbf{v}]$ with displacement fields $\mathbf{v} = [v_{\mathbf{x}}, v_{\mathbf{y}}]^T$ and $\mathcal{M}_i$ is the 4-neighborhood, this smoothness term becomes the first order derivative of the optical flow as in many conventional methods [2], [22], [23], [61]. However, non-rigid deformations frequently occur 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 affine transformations $\mathbf{T}_i$ in a moving least squares manner [15] 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 $\mathbf{T}_u \mathbf{u}$. Unlike conventional moving least squares solvers [15], [58], our smoothness term incorporates an adaptive weight υ^I_{iu} guided by the image

I as in [59], [60] for image structure-aware regularization, defined similar to ω_{ij}^I.

3.1.3 Overview

Minimizing the energy function $E(\mathbf{T})$ in (2) is a non-convex optimization problem defined over an infinite continuous solution space. With fine-scale discretization of this space, affine transformation fields could be estimated through discrete global optimization [18], [20], but at a tremendous computational cost. Furthermore, due to the difficulty of linearizing the non-convex data term, conventional continuous optimization techniques [62], [63], [64] also cannot be applied directly. We instead use a penalty decomposition scheme to alternately solve for the discrete and continuous affine transformation fields. An efficient filter-based discrete local optimization technique is used to solve the non-convex data term and locally estimate discrete affine transformations in a manner similar to PMF [24]. The weakness of the implicit smoothness constraint 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 dense semantic correspondence under spatially-varying affine fields, we extend the FCSS descriptor [11] by reformulating the sampling patterns.

3.2 Affine-FCSS Descriptor

To estimate a matching cost, a dense descriptor $\mathcal{D}_i$ should be extracted over the local support window of each image point i. For this we employ the state-of-the-art FCSS descriptor [11] for dense semantic correspondence, which formulates LSS [45] 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 $\mathcal{D}_i = \{\mathcal{D}_i^l\}$ for $l = \{1, ..., L\}$ with the maximum number of sampling patterns L, where the feature values are computed as

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

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

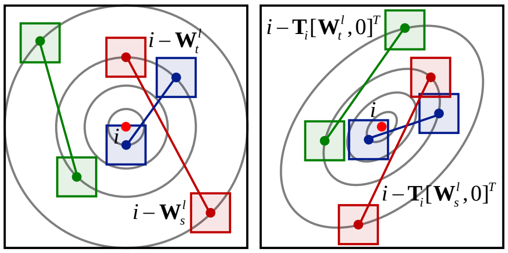
$$\mathcal{S}(i - \mathbf{W}_s^l, i - \mathbf{W}_t^l) = \|\mathcal{F}(\mathbf{A}_i; \mathbf{W}_s^l) - \mathcal{F}(\mathbf{A}_i; \mathbf{W}_t^l)\|^2, \tag{6}$$

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 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 to account for the affine transformations. To expedite this computation, we first pre-compute $\mathbf{A}$ over the entire image domain by passing I 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

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

With this approach, repeated computation of convolutional activations over different affine transformations can be avoided. It should be noted that for full affine invariance, the receptive fields for measuring self-similarity should also be transformed. However, transforming only the sampling patterns without transforming the receptive fields, as done in [65], [66], can nevertheless be effective in dealing with geometric variations. Differences between the FCSS descriptor and the affine-FCSS descriptor are illustrated in Fig. 2.


$i-\mathbf{W}_t^l$
i
$i-\mathbf{W}_s^l$
$i-\mathbf{T}_i[\mathbf{W}_t^l,0]^T$
i
$i-\mathbf{T}_i[\mathbf{W}_s^l,0]^T$

(a) FCSS [11] (b) Affine-FCSS

Fig. 2. Illustration of (a) FCSS descriptor [11] 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 transformation $\mathbf{T}_i$.

3.3 Solution

Since affine transformation fields $\mathbf{T}$ are defined in an infinite continuous solution space, minimizing the energy function $E(\mathbf{T})$ in (2) directly is infeasible. To solve this, we cast the inference of dense affine transformation fields as a discrete label optimization problem with continuous regularization. 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:

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

Since this energy function is based on two affine transformations, $\mathbf{T}$ and $\mathbf{L}$, we employ alternating minimization to solve for them and boost matching performance in a synergistic manner. We split the optimization of $E_{\text{aux}}(\mathbf{T}, \mathbf{L})$ 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 to infinity through the iterations drives the affine fields $\mathbf{T}$ and $\mathbf{L}$ together and eventually results in $\lim_{\mu \to \infty} E_{\text{aux}} \approx E$.

3.3.1 Discrete Local Optimization

To infer the discrete affine transformation field $\mathbf{T}^t$ with $\mathbf{L}^{t-1}$ being fixed at the t-th iteration, we reformulate the energy

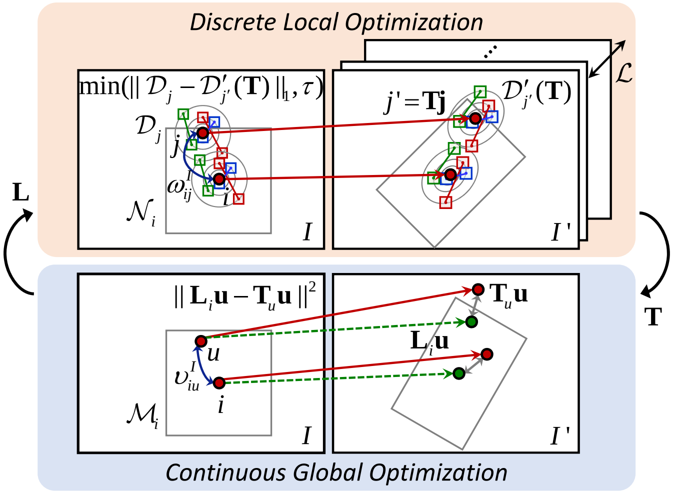

Discrete Local Optimization
$\min(\|\mathcal{D}_j - \mathcal{D}'_{j'}(\mathbf{T})\|_1, \tau)$
$\mathcal{D}_j$
j
ω^I_{ij}
$\mathcal{N}_i$
i
I
$j' = \mathbf{T}j$
$\mathcal{D}'_{j'}(\mathbf{T})$
I'
$\mathcal{L}$
$\mathbf{L}$
$\mathbf{T}$
$\|\mathbf{L}_i\mathbf{u} - \mathbf{T}_u\mathbf{u}\|^2$
$\mathbf{T}_u\mathbf{u}$
$\mathbf{L}_i\mathbf{u}$
u
υ^I_{iu}
i
$\mathcal{M}_i$
I
I'
Continuous Global Optimization

Fig. 3. Our DCTM method consists of discrete optimization and continuous optimization. Our DCTM method differs from the conventional PMF [24] by alternately optimizing the discrete label space and updating the discrete labels through continuous regularization.

function in (8) as follows:

$$\sum_i \sum_{j\in\mathcal{N}_i} \omega^I_{ij} \min(\|\mathcal{D}_j - \mathcal{D}'_{j'}(\mathbf{T}_i)\|_1, \tau) + \mu^t \sum_i \|\mathbf{T}_i - \mathbf{L}^{t-1}_i\|^2 + \lambda \sum_i \sum_{u\in\mathcal{M}_i} \upsilon^I_{iu} \|\mathbf{T}_u\mathbf{u} - \mathbf{L}^{t-1}_i\mathbf{u}\|^2. \tag{9}$$

To solve this, we first discretize the continuous parameter space and then solve the problem through filter-based label inference such as PMF [24]. For the affine field $\mathbf{T}$ within the set of discrete affine transformation candidates $\mathcal{L}$, the matching cost between descriptors $\mathcal{D}_j$ and $\mathcal{D}'_{j'}(\mathbf{T})$ is first measured as

$$C_j(\mathbf{T}) = \min(\|\mathcal{D}_j - \mathcal{D}'_{j'}(\mathbf{T})\|_1, \tau), \tag{10}$$

where $\mathcal{D}'_{j'}(\mathbf{T})$ is the affine-varying descriptor with respect to $\mathbf{T}$. 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 aggregation.

To aggregate the raw matching costs efficiently, we apply edge-aware filtering (EAF) on $C_i(\mathbf{T})$ such that

$$\bar{C}_i(\mathbf{T}) = \sum\nolimits_{j\in\mathcal{N}_i} \omega^I_{ij} C_j(\mathbf{T}), \tag{11}$$

where ω^I_{ij} is the adaptive weight of a support pixel j, which can be defined in various ways with respect to the structures of the image I [59], [60], [67]. Note that a simplified version of affine-invariant cost aggregation along an image row has been used in the context of stereo matching [22], [23], [24] and has shown state-of-the-art performance.

In determining the affine field $\mathbf{T}$, the matching costs additionally account for the previously estimated affine transformation field $\mathbf{L}^{t-1}$ through the following term:

$$G_i(\mathbf{T}) = \mu^t\|\mathbf{T} - \mathbf{L}^{t-1}_i\|^2 + \lambda \sum\nolimits_{u\in\mathcal{M}_i} \upsilon^I_{iu}\|\mathbf{T}\mathbf{u} - \mathbf{L}^{t-1}_i\mathbf{u}\|^2. \tag{12}$$

Since $\|\mathbf{T}\mathbf{u} - \mathbf{L}^{t-1}_i\mathbf{u}\|^2 = \|(\mathbf{T} - \mathbf{L}^{t-1}_i)\mathbf{u}\|^2$ and $\mathbf{T} - \mathbf{L}^{t-1}_i$ is independent of pixel u within the support window $\mathcal{M}_i$, $G_i(\mathbf{T})$ also can be efficiently computed by using fast EAFs [25], [68] with marginal computation overhead for varying $\mathbf{T}$ within the set of discrete affine transformation candidates $\mathcal{L}$.

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

$$\mathbf{T}^t_i = \mathrm{argmin}_{\mathbf{T}\in\mathcal{L}} \left(\bar{C}_i(\mathbf{T}) + G_i(\mathbf{T})\right). \tag{13}$$

Superpixel-based Iterative Inference: In filter-based discrete local optimization in (13), exhaustively evaluating the aggregated costs for every label $\mathcal{L}$ is still prohibitively time-consuming. A fast randomized search by PatchMatch [26] could be used to reduce computational complexity with respect to the search space, but its weak implicit smoothness constraint makes it more susceptible to erroneous local minima in a high dimensional label space such as for affine transformations. Additionally, a fragmented label search used in PatchMatch hinders the application of constant-time EAFs [24] for efficiently computing the aggregated cost in (11). So we follow the key idea of PMF [24] which uses segments or superpixels [69] to synergistically leverage the cost filtering and randomized search of PatchMatch [26]. Superpixels are utilized as the basic units for performing label propagation, randomized search, and subimage-based efficient cost aggregation collaboratively. However, our optimization differs from PMF [24] by optimizing the discrete label space with continuous regularization during the iterations, which facilitates convergence and boosts matching accuracy.

Specifically, we first decompose an image I into a set of K disjoint segments $\{S(k)\}$ for $k = \{1, ..., K\}$ 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 at odd iterations and reverse scan order at even iterations. 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 $\mathcal{L}_{\mathrm{prop}}$ is determined for i. For these $\mathcal{L}_{\mathrm{prop}}$, constant-time EAF-based cost aggregation is then performed [60] for the segment $S(k)$. In the random search step, a center-biased random search as done in PMF [24] is performed for the current segment $S(k)$. For the random search, the possible affine transformations $\Delta\mathbf{T}$ are set as a combination of translations in the x- and y-directions $[-h, h]$, $[-w, w]$ (where h and w are the height and width of the image, respectively), scales in the x- and y-directions $[1/2, 2]$, $[1/2, 2]$, rotation about the origin with the angle $[-\pi/2, \pi/2]$, shear transformation in the x- and y-directions with the angle $[-\pi/2, \pi/2]$, and reflection about the origin, x- and y-directions. By evaluating a sequence of random labels $\mathcal{L}_{\mathrm{rand}}$ sampled around the current best label $\mathbf{T}^*$, i.e., $\mathbf{T}^* + 0.5^l\Delta\mathbf{T}$ for $l = \{0, ..., |\mathcal{L}_{\mathrm{rand}}|\}$ as in PMF [24], the current best affine transformation fields are determined. After an iteration of the *propagation* and *random search* steps for all segments, we apply continuous optimization as described in the following section to regularize the discrete affine transformation fields.

3.3.2 Continuous Global Optimization

To solve the continuous affine transformation field $\mathbf{L}^t$ with $\mathbf{T}^t$ being fixed, we formulate the problem as an affine trans-

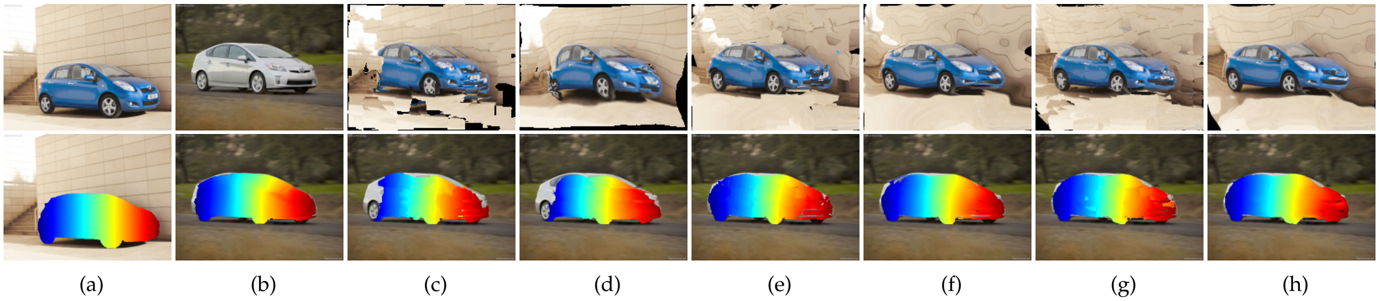


Fig. 4. DCTM convergence: (a) Source image; (b) Target image; Iterative evolution of warped images (c), (e), (g) after discrete optimization and (d), (f), (h) after continuous optimization after iteration 1, 2, and 3. Our DCTM optimizes the discrete label space with continuous regularization during the iterations, which facilitates convergence and boosts matching performance.

formation field regularization with the following energy:

$$\sum_i \left(\mu^t \|\mathbf{L}_i - \mathbf{T}_i^t\|^2 + \lambda \sum_{u\in\mathcal{M}_i} \upsilon_{iu}^I \|\mathbf{L}_i\mathbf{u} - \mathbf{T}_u^t\mathbf{u}\|^2 \right). \tag{14}$$

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 [15] at the cost of quantization errors or parallel processing [58] with an additional hardware. In contrast, our method optimizes the objective with a sparse matrix solver, yielding a substantial runtime gain. Since $\mathbf{L}_i$ can be formulated in the $\mathbf{x}$- and $\mathbf{y}$-directions separatively, i.e., $\mathbf{L}_{i,\mathbf{x}}$ and $\mathbf{L}_{i,\mathbf{y}}$, we decompose the objective into two separable energy functions. For the $\mathbf{x}$-direction, the energy function can be represented as

$$\sum_i \left(\mu^t \|\mathbf{L}_{i,\mathbf{x}} - \mathbf{T}_{i,\mathbf{x}}^t\|^2 + \lambda \sum_{u\in\mathcal{M}_i} \upsilon_{iu}^I \|\mathbf{L}_{i,\mathbf{x}}\mathbf{u} - \mathbf{T}_{u,\mathbf{x}}^t\mathbf{u}\|^2 \right). \tag{15}$$

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

$$(\mu^t/\lambda\mathbf{I} + \mathbf{U}^I)\mathbf{L}_\mathbf{x}^t = (\mu^t/\lambda\mathbf{I} + \mathbf{K}^I)\mathbf{T}_\mathbf{x}^t, \tag{16}$$

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

$$\mathbf{U}^I = \begin{bmatrix} \psi(\mathbf{V}^I X^2) & \psi(\mathbf{V}^I XY) & \psi(\mathbf{V}^I X) \\ \psi(\mathbf{V}^I XY) & \psi(\mathbf{V}^I Y^2) & \psi(\mathbf{V}^I Y) \\ \psi(\mathbf{V}^I X) & \psi(\mathbf{V}^I Y) & \mathbf{I}_{N\times N} \end{bmatrix}, \tag{17}$$

and

$$\mathbf{K}^I = \begin{bmatrix} \mathbf{V}^I\psi(X^2) & \mathbf{V}^I\psi(XY) & \mathbf{V}^I\psi(X) \\ \mathbf{V}^I\psi(XY) & \mathbf{V}^I\psi(Y^2) & \mathbf{V}^I\psi(Y) \\ \mathbf{V}^I\psi(X) & \mathbf{V}^I\psi(Y) & \mathbf{V}^I \end{bmatrix}, \tag{18}$$

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

The final result $\mathbf{L}_\mathbf{x}^t$ is then written as follows:

$$\mathbf{L}_\mathbf{x}^t = (\mu^t/\lambda\mathbf{I} + \mathbf{U}^I)^{-1}(\mu^t/\lambda\mathbf{I} + \mathbf{K}^I)\mathbf{T}_\mathbf{x}^t. \tag{19}$$

Algorithm 1: DCTM Framework

```
Input: images $I$, $I'$, descriptor network parameter $\mathbf{W}$
Output: dense affine transformation fields $\mathbf{T}$
Parameters: number of segments $K$, pyramid levels $M$
      /* Initialization */
1 :   Partition $I$ into a set of disjoint $K$ segments $\{S(k)\}$
2 :   Initialize affine fields as $\mathbf{L}^{\{0\}} = [\mathbf{I}_{2\times 2}, \mathbf{0}_{2\times 1}]$
      for $m = 1 : M$ do
3 :       Build $\mathbf{A}^{\{m\}}$, $\mathbf{A}'^{,\{m\}}$ for $I^{\{m\}}$, $I'^{,\{m\}}$
4 :       Initialize affine fields $\mathbf{T}^{\{m\}} = \mathbf{L}^{\{m-1\}}$
5 :       Compute $\mathcal{D}$ using $\mathbf{A}^{\{m\}}$
          while not converged do
              /* Discrete Local Optimization */
6 :           Initialize affine fields $\mathbf{T}^t = \mathbf{L}^{t-1}$
              for $k = 1 : K$ do
                  /* Propagation */
7 :               For $S(k)$, construct affine candidates $\mathbf{T} \in \mathcal{L}_{\mathrm{prop}}$
                   from neighboring segments
8 :               For $\mathbf{T}$, compute affine-varying $\mathcal{D}'(\mathbf{T})$ using $\mathbf{A}'^{,\{m\}}$
9 :               Build cost volumes $\bar{C}(\mathbf{T})$ and $G(\mathbf{T})$
10 :              Determine $\mathbf{T}^t$ using (13)
                  /* Random Search */
11 :              Construct affine candidates $\mathbf{T} \in \mathcal{L}_{\mathrm{rand}}$
                   from randomly sampled affine fields
12 :              Determine $\mathbf{T}^t$ by Step 8-10
              end for
              /* Continuous Global Optimization */
13 :          Estimate affine fields $\mathbf{L}^t$ from $\mathbf{T}^t$ using (19)
14 :          Enlarge $\mu$ such that $\mu^{t+1} = c\mu^t$
          end while
      end for
```

Since υ_{iu}^I is the adaptive weight, the matrices $\mathbf{U}^I$ and $\mathbf{K}^I$ can be efficiently computed using fast EAFs [60], [67]. Furthermore, since $\mu/\lambda\mathbf{I} + \mathbf{U}^I$ is a block-diagonal matrix, $\mathbf{L}_\mathbf{x}^t$ can be estimated efficiently using a fast sparse matrix solver [70]. After optimizing $\mathbf{L}_\mathbf{y}^t$ in a similar manner, we then have the continuous affine fields $\mathbf{L}^t$.

After each iteration, we enlarge μ such that $\mu^{t+1} = c\mu^t$ with a constant value $1 < c \leq 2$ to accelerate convergence. Fig. 3 summarizes our DCTM method, and Fig. 4 illustrates the convergence of our DCTM method.

3.3.3 Coarse-to-Fine Inference

Although our basic matching framework estimates reliable affine fields, it may exhibit limited performance on weakly- or repeated-textured regions. To alleviate these limitations, we employ a coarse-to-fine approach to boost matching performance and convergence based on the observation that correspondences estimated at a coarse image scale tend to be more reliable for weakly-textured regions, while correspondences estimated at a fine scale localize and preserve structure and motion details much better.

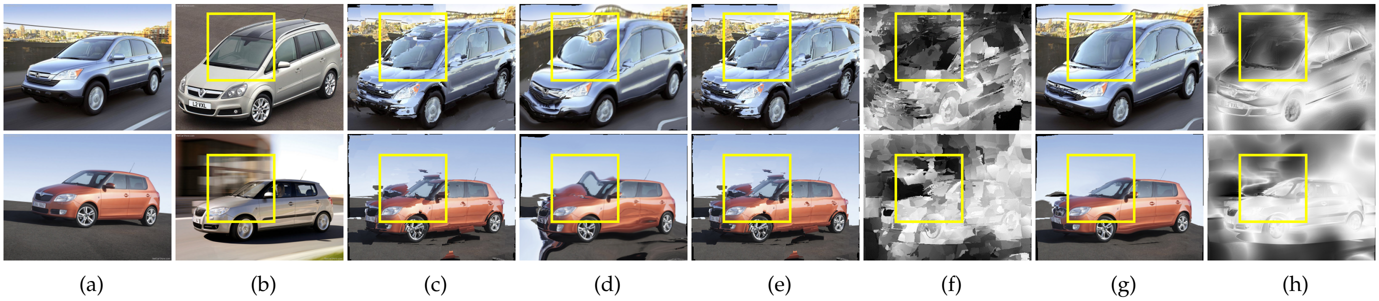


Fig. 5. Comparison of DCTM and CC-DCTM: (a), (b) source and target images, warped images with estimated correspondences after discrete and continuous optimization in (c), (d) DCTM, and (e), (g) CC-DCTM, and (f), (h) their corresponding confidence maps. The estimated confidences in CC-DCTM effectively reduce the effects of outliers during an iteration (top) and alleviate the ambiguity between image structures and correspondence fields (bottom), which greatly improves matching quality and convergence.

Specifically, images $I^{\{m\}}$ are constructed at M image pyramid levels $m = \{1, ..., M\}$ and affine transformation fields $\mathbf{T}^{\{m\}}$ and $\mathbf{L}^{\{m\}}$ are predicted at level m. Coarser scale results are then used as initialization for the finer levels. Furthermore, at the coarsest level, an unconstrained *random search* is conducted for possible affine transformation fields (i.e., for search space $\mathbf{T}^* + 0.5^l \Delta\mathbf{T}$). However, at the finer levels, the search range in the *random search* is increasingly constrained to avoid erroneous local minima (i.e., for search space $\mathbf{T}^* + r0.5^l \Delta\mathbf{T}$ with constraint r). Algorithm 1 summarizes the overall procedure of our DCTM method.

3.4 Extension of DCTM

For semantically similar images, there frequently exist inherently unmatchable regions caused by large viewpoint changes, non-rigid deformations, noise, or severe appearance variations [10], [29]. As in other algorithms, feature descriptors inherently fail to capture reliable matching evidence on such regions, and this problem often inhibits the convergence of DCTM to a global minimum. Moreover, cost aggregation guided only by the image I cannot fully estimate a transformation field in a structure-adaptive manner when there is inconsistency between structures of the image and correspondence fields [31], [61]. To better deal with such effects and improve convergence, we propose a correspondence constrained-DCTM, denoted as CC-DCTM, which leverages correspondence consistency between source and target images to detect occlusions and outliers, and incorporates a correspondence-aware cost aggregation and regularization schemes, as exemplified in Fig. 5.

3.4.1 Model

We reformulate our energy function to reliably aggregate and regulate the affine transformation fields by using only confident pixels within a local neighborhood. To this end, the confident adaptive weights are defined as

$$\bar{\omega}^J_{ij} \propto \omega^J_{ij}\rho_j, \quad \bar{\upsilon}^J_{iu} \propto \upsilon^J_{iu}\rho_u, \tag{20}$$

where ω^J_{ij} and υ^J_{iu} represent adaptive weights, defined similar to ω^I_{ij} and υ^I_{iu}, using guidance J with the image I as static guidance and the affine field $\mathbf{T}$ as dynamic guidance, an approach that has shown reliable performance in [71]. This static and dynamic guidance involves computation over a range with respect to a 7D vector[1] when applying the confidence-guided edge-aware filtering, which significantly increases the computational burden needed for employing constant-time EAFs [60], [67]. To alleviate this problem, we first apply principal components analysis (PCA) to project the 7D vector into a 1D vector for dimension reduction [72], [73], and then apply constant-time EAFs [60], [67] on this guidance image J.

```
Algorithm 2: CC-DCTM Framework
Input: images $I$, $I'$, descriptor network parameter $\mathbf{W}$
Output: dense affine transformation fields $\mathbf{T}$
Parameters: number of segments $K$, pyramid levels $M$
     /* Initialization */
1 :  Partition $I$, $I'$ into a set of disjoint $K$ segments $\{S(k)\}$, $\{S'(k)\}$
2 :  Initialize $\mathbf{L}^{\{0\}} = [\mathbf{I}_{2\times2}, \mathbf{0}_{2\times1}]$, $\mathbf{L}'^{,\{0\}} = [\mathbf{I}_{2\times2}, \mathbf{0}_{2\times1}]$
     for $m = 1 : M$ do
3 :      Build $\mathbf{A}^{\{m\}}$, $\mathbf{A}'^{,\{m\}}$ for $I^{\{m\}}$, $I'^{,\{m\}}$
4 :      Initialize $\mathbf{T}^{\{m\}} = \mathbf{L}^{\{m-1\}}$, $\mathbf{T}'^{,\{m\}} = \mathbf{L}'^{,\{m-1\}}$
         while not converged do
             /* Discrete Local Optimization */
5 :          Estimate $\mathbf{T}^t$, $\mathbf{T}'^{,t}$ from $\mathbf{L}^{t-1}$, $\mathbf{L}'^{,t-1}$
               through Step 7-12 in Algorithm 1.
6 :          Compute confidence $\rho$, $\rho'$ of $\mathbf{T}^t$, $\mathbf{T}'^{,t}$ using (21)
             /* Continuous Global Optimization */
7 :          Estimate affine fields $\mathbf{L}^t$, $\mathbf{L}'^{,t}$ from $\mathbf{T}^t$, $\mathbf{T}'^{,t}$ using (16)
8 :          Compute confidence $\rho$, $\rho'$ of $\mathbf{L}^{t-1}$, $\mathbf{L}'^{,t-1}$ using (21)
         end while
     end for
```

The confidence ρ_i is defined as follows:

$$\rho_i = \exp(-\|i + \mathbf{T}'_{i'}\mathbf{i}'\|_1/\sigma), \tag{21}$$

where σ represents the parameter for the Gaussian kernel. It is designed to encode the confidence of affine transformation field $\mathbf{T}_i$ by checking the consistency between pixel i in I and a bi-directional mapping $\mathbf{T}'_{i'}\mathbf{i}'$ for $i' = \mathbf{T}_i\mathbf{i}$ in I'. Thus, the confidence works in such a way that the matching costs of unreliable nearby points are excluded from the aggregation in a correspondence-aware manner. It should be noted that correspondence consistency has been popularly used to eliminate erroneous correspondences as a post-processing step [24], [29], [74]. Unlike these, our method incorporates this into the iterative optimization framework. This enables actively detection and handling of occlusion regions and outliers, where feature descriptors frequently fail to capture reliable matching evidence. Fig. 5 visualizes confidence maps formed in CC-DCTM.

3.4.2 Solution

Minimizing the energy function may be hard since the confident adaptive weights $\bar{\omega}^J$ and $\bar{\upsilon}^J$ need to be dynamically defined with respect to the estimated affine transformation

[1] The 7D vector is composed of 1D for image I and 6D for the vector form of affine transformation fields $\mathbf{T}$.

field $\mathbf{T}$. Fortunately, the penalty decomposition scheme for DCTM in (8), which alternatively solves for the discrete and continuous optimization, remains effective for minimizing this energy function. Concretely, when solving the discrete optimization at the t-th iteration, the confidence ρ is determined with respect to $\mathbf{L}^{t-1}$, while it is determined with respect to $\mathbf{T}^t$ when solving the continuous optimization.

In the discrete optimization, the edge-aware aggregation in (11) can be defined as

$$\sum\nolimits_{j\in\mathcal{N}_i}\bar{\omega}^J_{ij}C_j(\mathbf{T}) = \sum\nolimits_{j\in\mathcal{N}_i}\omega^J_{ij}\rho_j C_j(\mathbf{T}) / \sum\nolimits_{j\in\mathcal{N}_i}\omega^J_{ij}\rho_j, \tag{22}$$

which can be computed efficiently by applying the constant-time EAF to $\rho_i C_i(\mathbf{T})$ and ρ_i, respectively, with guidance J for the image I and current affine fields $\mathbf{L}^{t-1}$, similar to [67], [75]. In a similar way, $\bar{G}_i(\mathbf{T})$ in (12) can also be computed efficiently.

In the continuous optimization, $\mathbf{U}^I$ and $\mathbf{K}^I$ in (16) also can be efficiently determined using a computational scheme similar to that for (22), which is used to obtain $\bar{\mathbf{U}}^J$ and $\bar{\mathbf{K}}^J$. For example, similar to [61], $\bar{\mathbf{V}}^J X$ can be solved as

$$\bar{\mathbf{V}}^J X = \mathbf{V}^J \psi(\mathbf{F}) X \oslash \mathbf{V}^J \mathbf{F}, \tag{23}$$

where $\oslash$ represents the element-wise division operator. $\mathbf{F}$ is the matrix form of ρ_i for all i. $\mathbf{V}^J$ is a kernel function whose nonzero elements are given by the confidence-guided edge-aware weights with the guidance J of the image I and current affine fields $\mathbf{T}^t$. Thus, it remains efficiently computable. In a similar manner, $\bar{\mathbf{V}}^J Y$, $\bar{\mathbf{V}}^J X^2$, $\bar{\mathbf{V}}^J Y^2$, $\bar{\mathbf{V}}^J XY$, $\bar{\mathbf{V}}^J \psi(X)$, and $\bar{\mathbf{V}}^J \psi(Y)$ can be efficiently computed, and they are used to solve (19). Algorithm 2 provides a summary of the CC-DCTM optimization.

4 Experimental Results

4.1 Experimental Settings

For our experiments, we use the FCSS descriptor learned on a version of the Caltech-101 dataset [34] that excludes image pairs used for testing, without further training. The EAF for ω^I_{ij}, υ^I_{iu}, ω^J_{ij}, and υ^J_{iu} are performed using the guided filter [68] because of its robustness and efficiency, where the radius and smoothness parameters are set to $\{16, 0.01\}$. It should be noted that any other features and EAFs could be used in our approach. The weights in the energy function are set initially to $\{\lambda, \mu\} = \{0.01, 0.1\}$ by cross-validation, and μ is increased by factor $c = 1.8$ with subsequent iterations. τ and σ are set to 4 and 30, respectively. The constraint r for random search is set to 0.3. The SLIC [69] algorithm is used for superpixel segmentation and the segment number K is set to increase sublinearly with an image size, e.g., $K = 500$ for 640×480 images, by considering the trade-off between efficiency and robustness (see [7], [24]). The image pyramid level M is set to 3. In experiments, estimated pixel-varying affine transformation fields with DCTM and CC-DCTM are represented as displacement vectors (i.e., flow fields).

In the following, we comprehensively evaluate DCTM and CC-DCTM through comparisons to state-of-the-art methods for semantic correspondence, including SF [2], DSP [37], Zhou et al. [10], Taniai et al. [29], PF [30], Ufer et al. [46], OHG [39], ANet [43], Deep Image Analogy [76], and the SF optimizer[2] with DSP-SIFT [77], VGG[3] [78], UCN [9], and FCSS [11] descriptor. Furthermore, geometric-invariant methods including SLS [12], SSF [13], SegSIFT [80], Lin et al. [17], DFF [7], GDSP [14], and GMat [50] were also evaluated. Performance is measured on the TSS benchmark [29], PF-WILLOW dataset [30], PF-PASCAL dataset [31], CUB-200-2011 dataset [32], PASCAL-VOC dataset [33], and Caltech-101 benchmark [34]. To validate the components of DCTM, we examine the effects of dropping the continuous optimization (wo/Cont.) and the coarse-to-fine scheme (wo/C2F). To validate the components of CC-DCTM, we also observe the results from removing the correspondence consistency (wo/CC) (i.e., $\rho = 1$) and the confidence-guided EAF (wo/CEF) (i.e., $J = I$).

In Sec. 4.2, we first analyze the convergence of DCTM and CC-DCTM. In Sec. 4.3, we then examine the performance of our methods compared to other matching methods when combined with other descriptors. We then evaluate our matching results compared to the state-of-the-art methods on various benchmarks in Sec. 4.4. We finally evaluate the computation speed in Sec. 4.5.

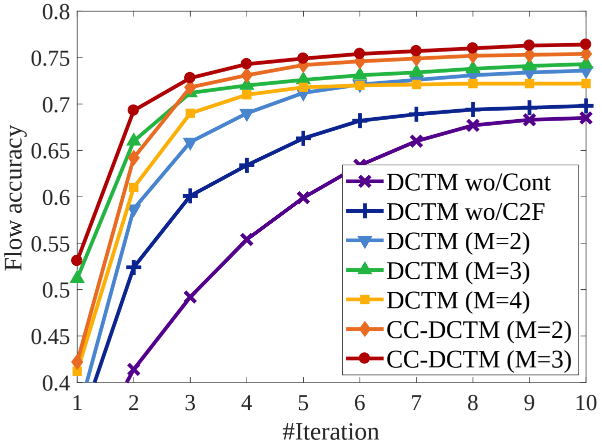


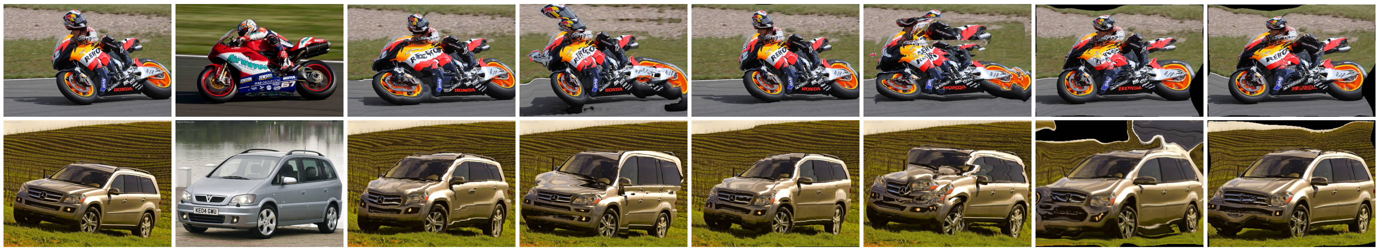
Fig. 6. Convergence analysis of DCTM and CC-DCTM on the TSS benchmark [29].

TABLE 1
Matching accuracy compared to state-of-the-art correspondence techniques with different features and matching methods on the TSS benchmark [29]. † denotes fine-tuned features.

Methods					
Features	Matching	FG3D	JODS	PASC.	Avg.
DAISY [27]	SF [2]	0.636	0.373	0.338	0.449
DSP-SIFT [77]		0.659	0.524	0.352	0.512
VGG [78]		0.756	0.490	0.360	0.535
FCSS [11]		0.830	0.653	0.494	0.660
VGG [78]	PF [30]	0.773	0.593	0.492	0.619
FCSS [11]		**0.839**	0.635	**0.582**	**0.685**
VGG† [78]	SCNet [47]	0.776	0.608	0.474	0.619
VGG† [78]	GMat [50]	0.835	**0.656**	0.527	0.673
DAISY [27]	DCTM	0.710	0.506	0.482	0.566
VGG [78]		0.790	0.611	0.528	0.630
FCSS [11]		**0.891**	**0.721**	**0.610**	**0.740**
FCSS [11]	CC-DCTM	**0.901**	**0.736**	**0.609**	**0.749**

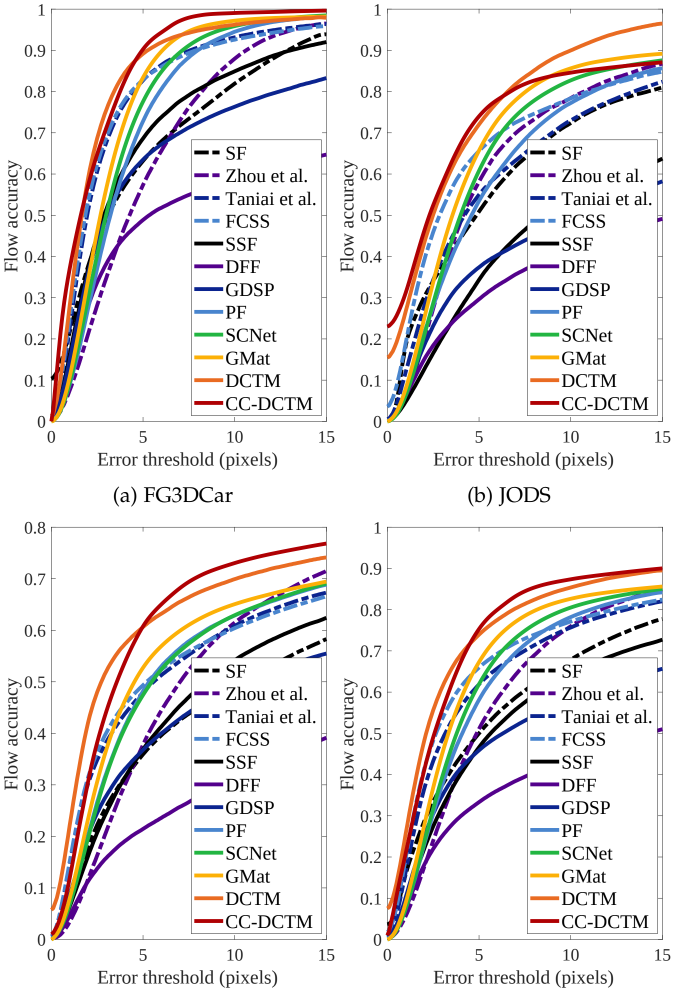
[2] For these experiments, we only utilized the optimizer used in SF, namely the hierarchical dual-layer belief propagation [2], with the alternative dense descriptors.

[3] In VGG, ImageNet pretrained VGG-Net [78] from the bottom conv1 to the conv3-4 layer was used with L_2 normalization [79].



(a) source image (b) target image (c) FCSS [11] (d) PF [30] (e) UCN [9] (f) SCNet [47] (g) DCTM (h) CC-DCTM

Fig. 7. Qualitative results on the TSS benchmark [29]. The source images were warped to the target images using correspondences.



(a) FG3DCar (b) JODS

(c) PASCAL (d) Average

Fig. 8. Average matching accuracy with respect to endpoint error threshold on the TSS benchmark [29].

4.2 Convergence Analysis

We first analyze the convergence of DCTM and CC-DCTM. For a quantitative analysis, we measure flow accuracy (described in the following subsection) on the TSS benchmark [29]. For each method, we measure the flow accuracy for different numbers of iterations up to the maximum number without early convergence termination. Fig. 6 shows the convergence analysis of DCTM and CC-DCTM for varying numbers of image pyramid levels M. The results of 'DCTM wo/Cont.' reveal the significance of continuous regularization in DCTM. With the continuous regularization, our methods converge in $3-5$ iterations. We also observe that matching quality and convergence speed are improved until $M = 3$ by enlarging the number of image pyramid levels, but using larger pyramid levels (e.g., $M = 4$) reduces matching accuracy due to greater ambiguity at the coarsest level. Based on these experiments, we set $M = 3$. Thanks particularly to the correspondence consistency and confidence-guided aggregation, CC-DCTM exhibits improved robustness and convergence compared to DCTM.

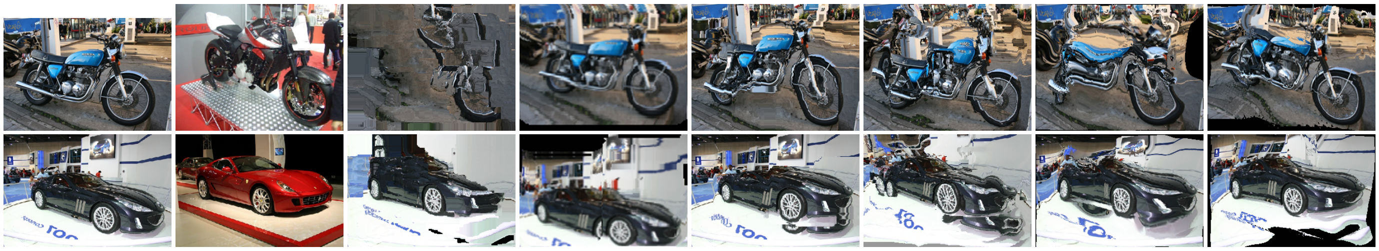
TABLE 2
Matching accuracy compared to state-of-the-art correspondence techniques on the TSS benchmark [29].

Methods	FG3D	JODS	PASC.	Avg.
SF [2]	0.632	0.509	0.360	0.500
DSP [37]	0.487	0.465	0.382	0.445
Zhou et al. [10]	0.721	0.514	0.436	0.556
Taniai et al. [29]	0.830	0.595	0.483	0.636
OHG [39]	0.875	0.708	**0.729**	**0.771**
SLS [12]	0.525	0.519	0.320	0.457
SSF [13]	0.687	0.344	0.370	0.467
SegSIFT [80]	0.612	0.421	0.331	0.457
Lin et al. [17]	0.406	0.283	0.161	0.283
DFF [7]	0.489	0.296	0.214	0.333
GDSP [14]	0.639	0.374	0.368	0.459
PF [30]	0.786	0.653	0.531	0.657
UCN [9]	0.853	0.672	0.511	0.679
FCSS [11]	0.830	0.653	0.494	0.660
GMat [50]	0.835	0.656	0.527	0.673
SCNet [47]	0.776	0.608	0.474	0.619
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
CC-DCTM wo/CC	0.883	0.716	0.607	0.735
CC-DCTM wo/CEF	**0.886**	**0.730**	**0.613**	**0.743**
CC-DCTM	**0.901**	**0.736**	0.609	**0.749**

4.3 Effects of Feature Descriptors

We then analyze the effects of feature descriptors in DCTM and CC-DCTM, and compare to other regularization or matching methods such as SF [2], PF [30], SCNet [47], and GMat [50] when combined with other descriptors[4] using DAISY [27], VGG [78], and FCSS [11]. Similar to Sec. 4.2, for a quantitative analysis, we measure flow accuracy on the Taniai benchmark [29]. Table 1 summarizes the state-of-the-art methods with their features and matching algorithms, and reports the matching accuracy. Matching methods with deep CNN-based features have shown improved performance over those with handcrafted features such as DSP-SIFT [77] and DAISY [27]. When comparing the perfor-

[4] These experiments use only the upright version of the descriptors since no techniques exist for computing the descriptors efficiently with respect to affine transformations.



(a) source image (b) target image (c) SSF [13] (d) GDSP [14] (e) FCSS [11] (f) SCNet [47] (g) DCTM (h) CC-DCTM

Fig. 9. Qualitative results on the PF-WILLOW benchmark [30]. The source images were warped to the target images using correspondences.

mance with VGG [78], DCTM shows the state-of-the-art performance except for GMat [50]. Note that GMat [50] requires substantial additional training of CNNs for features and regularizations, but DCTM is training-free and can effectively handle geometric variations in more challenging cases, which will be shown in the following experiments. By using a strong feature such as FCSS [11] for semantic correspondence, the performance of DCTM can be boosted, as in SF [2] and PF [30]. Moreover, thanks to the affine-varying features such as affine-FCSS, DCTM and CC-DCTM exhibit highly improved robustness and convergence.

4.4 Matching Results

4.4.1 Results on TSS Benchmark

We evaluate DCTM and CC-DCTM on the TSS benchmark [29], which consists of 400 image pairs divided into three groups: FG3DCar [81], JODS [82], and PASCAL [83]. As in [11], [29], flow accuracy was measured by computing the proportion of foreground pixels with an absolute flow endpoint error that is smaller than a threshold T, after resizing images so that its larger dimension is 100 pixels.

Fig. 7 displays qualitative results for dense flow estimation. Fig. 8 plots the flow accuracy with respect to error threshold T. Table 2 summarizes the matching accuracy for state-of-the-art correspondence techniques (for $T = 5$ pixels). Compared to methods based on handcrafted features [7], [13], [14], CNN-based methods [9], [11], [29], [47], [50] provide higher accuracy even though they do not consider geometric variations. Existing geometry-invariant methods [7], [13], [14], [17] cannot provide satisfactory performance when matching evidence is measured with handcrafted features. The method of Lin et al. [17] cannot estimate reliable correspondences due to unstable sparse correspondences. In contrast, our DCTM method provides state-of-the-art performance in most cases thanks to its discrete labeling optimization with continuous regularization and affine-FCSS, while OHG [46] shows state-of-the-art performance in some results. Furthermore, our CC-DCTM demonstrates improved convergence and state-of-the-art performance compared to the other methods. As shown in the results of 'CC-DCTM wo/CC' and 'CC-DCTM wo/CEF' in Table 2, the correspondence consistency and the confidence-guided edge-aware filtering clearly elevate matching accuracy.

TABLE 3
Matching accuracy compared to state-of-the-art correspondence techniques on the PF-WILLOW benchmark [30].

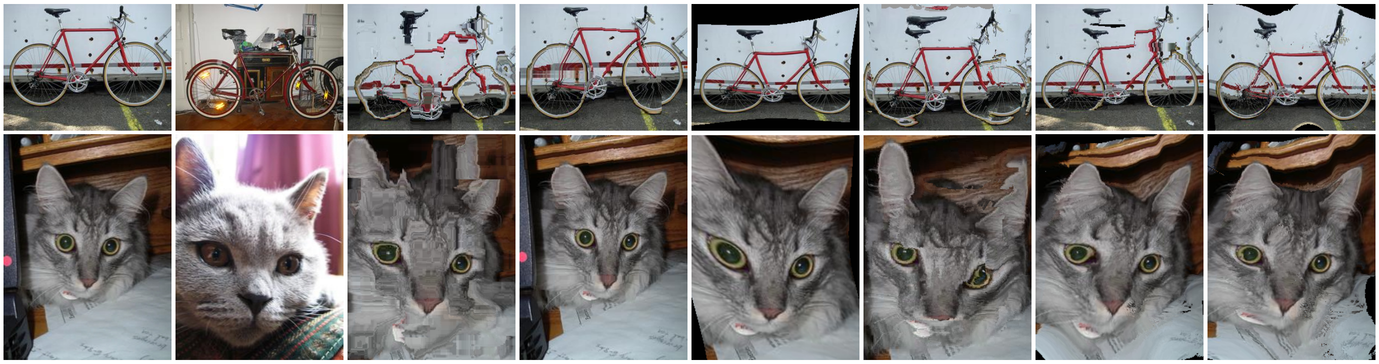
Methods	PCK		
	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.15$
Zhou et al. [10]	0.197	0.524	0.664
SSF [13]	0.292	0.401	0.531
Lin et al. [17]	0.192	0.354	0.487
DFF [7]	0.241	0.362	0.510
GDSP [14]	0.242	0.487	0.512
PF [30]	0.284	0.568	0.682
UCN [9]	0.241	0.540	0.665
FCSS [11]	0.354	0.532	0.681
GMat [50]	0.312	0.586	0.712
SCNet [47]	0.359	0.601	0.692
DCTM wo/Cont.	0.353	0.552	0.687
DCTM wo/C2F	0.368	0.568	0.702
DCTM	0.381	0.610	0.721
CC-DCTM wo/CC	**0.382**	**0.616**	**0.724**
CC-DCTM wo/CEF	**0.384**	**0.612**	**0.726**
CC-DCTM	**0.386**	**0.621**	**0.730**

TABLE 4
Matching accuracy compared to state-of-the-art correspondence techniques on the PF-PASCAL benchmark [31].

Methods	mIoU	PCK		
		$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.15$
Zhou *et al.* [10]	0.310	0.181	0.410	0.624
SSF [13]	0.297	0.210	0.382	0.511
Lin *et al.* [17]	0.279	0.204	0.368	0.498
DFF [7]	0.347	0.214	0.372	0.522
GDSP [14]	0.482	0.222	0.412	0.524
PF [30]	0.511	0.242	0.451	0.640
UCN [9]	0.502	0.241	0.493	0.621
FCSS [11]	0.591	**0.269**	0.459	**0.648**
GMat [50]	0.579	0.231	0.462	0.638
SCNet [47]	0.534	**0.264**	0.470	0.643
DCTM wo/Cont.	0.602	0.240	0.461	0.641
DCTM wo/C2F	0.610	0.243	**0.471**	0.642
DCTM	0.616	0.258	**0.476**	0.644
CC-DCTM wo/CC	**0.632**	0.259	0.472	0.640
CC-DCTM wo/CEF	**0.634**	0.263	0.470	**0.647**
CC-DCTM	**0.652**	**0.268**	**0.473**	**0.645**

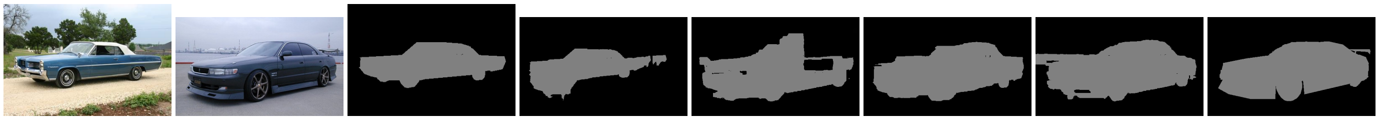
4.4.2 Results on PF-WILLOW Benchmark

We also evaluate our method on the PF-WILLOW benchmark [30], which includes 10 object sub-classes with 10 keypoint annotations for each image. For the evaluation metric, we use the probability of correct keypoint (PCK) between flow-warped keypoints and the ground truth [8], [30]. The warped keypoints are deemed to be correctly predicted if they lie within $\alpha \cdot \max(h_b, w_b)$ pixels of the ground-truth keypoints for $\alpha \in [0, 1]$, where h_b and w_b are the height and width of the object bounding box, respectively. The PCK values were measured for different correspondence techniques in Table 3. Fig. 9 shows qualitative results for dense flow estimation. Our DCTM method exhibits performance competitive to the state-of-the-art correspondence techniques. Our CC-DCTM method is especially effective in cases of severe appearance and shape variations compared



(a) source image (b) target image (c) VGG [78] (d) UCN [9] (e) GMat [50] (f) SCNet [47] (g) DCTM (h) CC-DCTM

Fig. 10. Qualitative results on the PF-PASCAL benchmark [31]. The source images were warped to the target images using correspondences.



(a) source image (b) target image (c) src. segments (d) FCSS [11] (e) SCNet [47] (f) DCTM (g) CC-DCTM (h) tar. segments

Fig. 11. Visualization of non-parametric semantic segmentation on the PF-PASCAL benchmark [31]. For visualization, color-coded source semantic segments were warped to the target images using correspondences.

to other methods.

4.4.3 Results on PF-PASCAL Benchmark

We evaluate DCTM and CC-DCTM on the PF-PASCAL benchmark [31], which contains 1,351 image pairs for 20 object categories with PASCAL keypoint annotations [84]. For the evaluation metric, we use the PCK between flow-warped keypoints and the ground truth as in the experiments on the PF-WILLOW benchmark [30]. Moreover, we also apply our methods to the non-parametric semantic segmentation task on the PF-PASCAL benchmark [31] in a manner where segmentation masks are transferred from source to target images using dense correspondences. For quantitative evaluation, we adopted the mean intersection over union (mIoU) between the predicted segmentations and ground truths.

The PCK values and mIoU are measured for different correspondence techniques in Table 4. Fig. 10 shows qualitative results for dense flow estimation. Fig. 11 shows the predicted semantic segmentation using dense correspondences. DCTM method exhibits outstanding performance compared to state-of-the-art dense correspondence estimation methods. CC-DCTM method again is found to be reliable especially under challenging correspondence settings.

4.4.4 Results on CUB-200-2011 Benchmark

We evaluate our DCTM and CC-DCTM on the CUB-200-2011 dataset [32], which contains 11,788 images of 200 bird categories, with 15 parts annotated. We follow the experimental configuration in [85], which utilizes 5,000 image pairs from the validation subset as testing pairs. For the evaluation metric, we use the PCK between flow-warped keypoints and the ground truth [85], where a match is considered correct if the predicted point is within $\alpha \cdot L_d$ of the mean diagonal length of the two images L_d.

The average PCK is measured for various descriptors and correspondence techniques in Table 5. Fig. 12 visualizes dense flow fields with keypoint annotation transfer. In this experiment, we evaluate descriptors including SIFT [36], VGG [78], and FCSS [11] using nearest neighbor (NN) search on uniformly sampled keypoints on the foreground with a stride of 8, following [85]. Our DCTM and CC-DCTM show competitive performance compared to methods such as DSP [37] and WarpNet [85].

TABLE 5
Matching accuracy compared to state-of-the-art correspondence techniques on the CUB-200-2011 benchmark [32].

Methods	Mean PCK		
	$\alpha = 0.01$	$\alpha = 0.05$	$\alpha = 0.10$
NN w/VGG [78]	0.113	0.501	0.620
NN w/FCSS [11]	**0.196**	**0.614**	**0.920**
DSP [37]	0.096	0.429	0.701
DSP w/VGG [78]	0.093	0.456	0.724
WarpNet [85]	0.121	0.602	0.814
DCTM	**0.212**	**0.657**	**0.924**
CC-DCTM	**0.245**	**0.668**	**0.892**

TABLE 6
Matching accuracy on the PASCAL-VOC dataset [33].

Methods	IoU	PCK	
		$\alpha = 0.05$	$\alpha = 0.1$
Zhou et al. [10]	-	-	0.24
UCN [9]	-	0.26	0.44
FCSS [11]	0.44	0.28	**0.47**
DSP w/ANet [43]	0.45	0.24	-
Deep Image Analogy [76]	-	-	0.21
DFF [7]	0.36	0.14	0.31
GDSP [14]	0.40	0.16	0.34
PF [30]	0.41	0.17	0.36
PF w/FCSS [11]	**0.46**	**0.29**	0.46
DCTM	**0.48**	**0.32**	**0.50**
CC-DCTM	**0.50**	**0.31**	**0.52**

4.4.5 Results on PASCAL-VOC Dataset

We conduct part segmentation experiments on the dataset provided by [41], where the images are sampled from the PASCAL-VOC parts dataset [33]. With human-annotated part segments, we measure part matching accuracy using the weighted intersection over union (IoU) score between transferred segments and ground truths, with weights deter-

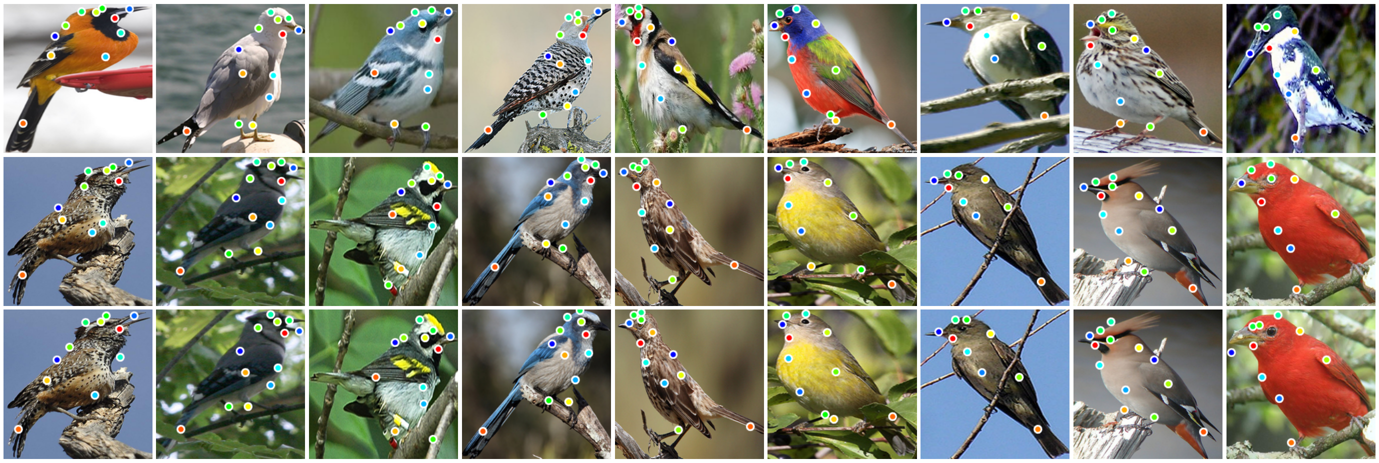
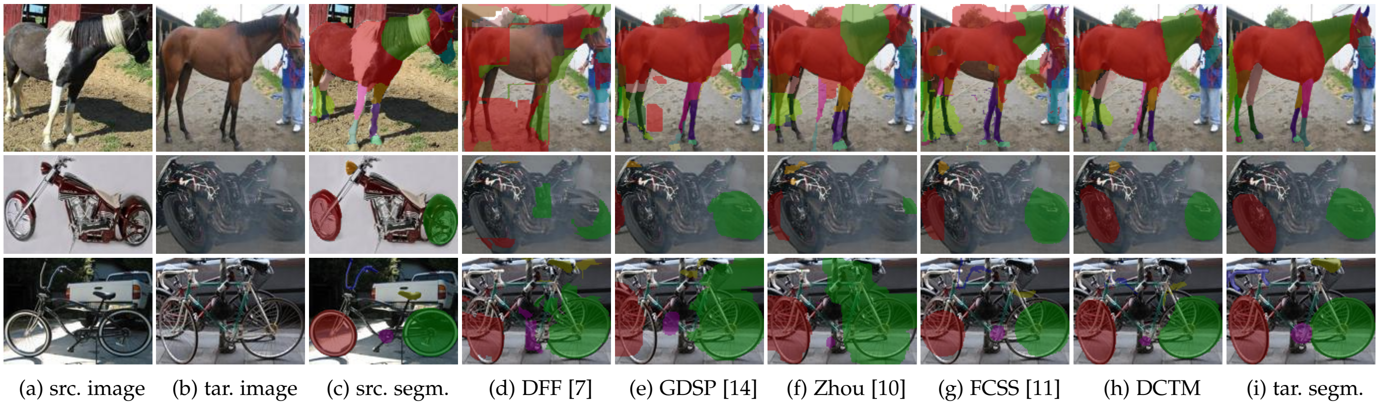
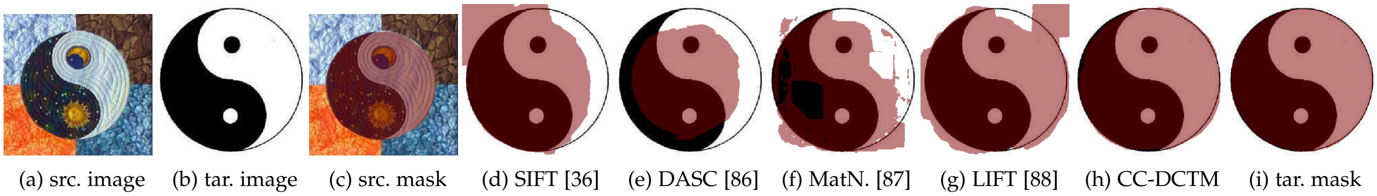


Fig. 12. Visualizations of dense flow fields with keypoint annotation transfer on the CUB-200-2011 benchmark [32]: (from top to bottom) source and target images with ground-truth keypoint annotations, and target images with warped keypoint annotations using correspondences from DCTM.



(a) src. image (b) tar. image (c) src. segm. (d) DFF [7] (e) GDSP [14] (f) Zhou [10] (g) FCSS [11] (h) DCTM (i) tar. segm.

Fig. 13. Visualizations of dense flow fields with color-coded part segments on the PASCAL-VOC part dataset [33]. The source part segments were warped to the target images using correspondences.



(a) src. image (b) tar. image (c) src. mask (d) SIFT [36] (e) DASC [86] (f) MatN. [87] (g) LIFT [88] (h) CC-DCTM (i) tar. mask

Fig. 14. Visualizations of dense flow fields with mask transfer on the Caltech-101 dataset [34]. The source masks were warped to the target images using correspondences.

TABLE 7
Matching accuracy on the Caltech-101 dataset [34].

Methods	LT-ACC	IoU	LOC-ERR
PF [30]	0.78	0.50	0.25
VGG [78]	0.78	0.51	0.25
OHG [39]	0.81	**0.55**	**0.19**
FCSS [11]	0.80	0.50	**0.21**
PF w/FCSS [11]	**0.83**	0.52	0.22
DCTM	**0.84**	**0.53**	**0.18**
CC-DCTM	**0.85**	**0.56**	**0.21**

mined by the pixel area of each part. To evaluate alignment accuracy, we measure the PCK metric using keypoint annotations for the 12 rigid PASCAL classes [89]. Table 6 summarizes the matching accuracy compared to state-of-the-art correspondence methods. Fig. 13 visualizes estimated dense flow with color-coded part segments. Our results are found to yield the highest matching accuracy.

4.4.6 Results on Caltech-101 Dataset

Our next experiments are on mask transfer using the Caltech-101 dataset [34]. Following the experimental protocol in [37], we randomly select 15 pairs of images for each object class, and evaluate the matching accuracy with three metrics: label transfer accuracy (LT-ACC) [3], the IoU metric, and the localization error (LOC-ERR) of corresponding pixel positions. Compared to the other benchmarks described above, the Caltech-101 dataset provides image pairs from a more diverse set of classes, enabling us to evaluate our method under more general correspondence settings. Table 7 summarizes the matching accuracy compared to the state-of-the-art correspondence methods. Fig. 14 visualizes estimated dense flow fields with mask transfer. Our DCTM and CC-DCTM clearly outperform the comparison techniques.

4.5 Computation Speed

In Fig. 15, we compare the computational speed of our methods to state-of-the-art methods. We implemented our

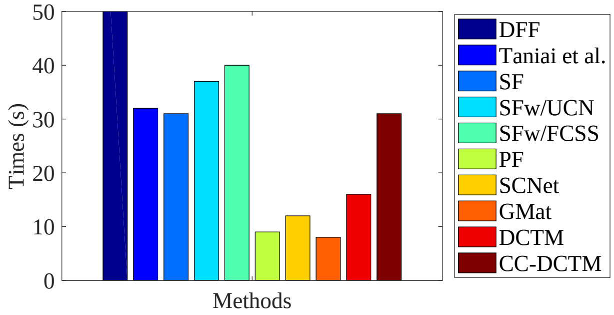


Fig. 15. Computation speed analysis of DCTM and CC-DCTM compared to other state-of-the-art methods for 320×240 images.

methods in Matlab/C++ on an Intel Core i7-3770 CPU at 3.40 GHz, and measured the runtime on a single CPU core. The computation time for CC-DCTM is higher than that of DCTM since it computes forward/backward affine fields for confidence computation. Even though our methods need more computation compared to some techniques, they exhibit clearly better matching performance.

5 Conclusion

We presented a novel method that 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 efficiently through constant-time edge-aware filtering and the affine-FCSS descriptor. The convergence and matching quality of DCTM are further elevated by leveraging correspondence consistency and confidence-guided edge-aware filtering. Further investigation may include examining how semantic correspondences computed from our methods could benefit single-image 3D reconstruction and instance-level object segmentation.

Acknowledgments

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (NRF-2017M3C4A7069370). This work of S. Kim was supported (in part) by the Yonsei University Research Fund (Yonsei Frontier Lab. Young Researcher Supporting Program) of 2018. This work of D. Min was supported in part by the Young Researcher Program through the NRF under Grant NRF-2018R1C1B6004622.

References

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

[2] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *IEEE Trans. PAMI*, vol. 33, no. 5, pp. 815–830, 2011.

[3] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing via label transfer," *IEEE Trans. PAMI*, vol. 33, no. 12, pp. 2368–2382, 2011.

[4] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *IJCV*, vol. 47, no. 1, pp. 7–42, 2002.

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

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

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

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

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

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

[11] 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.

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

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

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

[15] S. Schaefer, T. McPhail, and J. Warren, "Image deformation using moving least squares," *In: SIGGRAPH*, vol. 25, no. 3, pp. 533–540, 2006.

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

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

[18] Y. Boykov, O. Yeksler, and R. Zabih, "Fast approximation energy minimization via graph cuts," *IEEE Trans. PAMI*, vol. 23, no. 11, pp. 1222–1239, 2001.

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

[20] 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*, vol. 30, no. 6, pp. 1068–1080, 2008.

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

[22] F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz, "Pmbp: Patchmatch belief propagation for correspondence field estimation," *IJCV*, vol. 110, no. 1, pp. 2–13, 2014.

[23] 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.

[24] 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.

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

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

[27] E. Tola, V. Lenpetit, and P. Fua, "Daisy: An efficient dense descriptor applied to wide-baseline stereo," *IEEE Trans. PAMI*, vol. 32, no. 5, pp. 815–830, 2010.

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

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

[31] B. Ham, M. Cho, C. Schmid, and J. Ponce, "Proposal flow: Semantic correspondences from object proposals," *IEEE Trans. PAMI*, 2017.

[32] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," Tech. Rep., 2011.

[33] 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.

[34] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. PAMI*, vol. 28, no. 4, pp. 594–611, 2006.

[35] S. Kim, D. Min, S. Lin, and K. Sohn, "Dctm: Discrete-continuous transformation matching for semantic flow," *In: ICCV*, 2017.

[36] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

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

[38] T. Hassner, "Viewing real-world faces in 3d," *In: ICCV*, 2013.

[39] F. Yang, X. Li, H. Cheng, J. Li, and L. Chen, "Object-aware dense semantic correspondence," *In: CVPR*, 2017.

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

[41] 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.

[42] Online., http://www.shapenet.org/.

[43] D. Novotny, D. Larlus, and A. Vedaldi, "Anchornet: A weakly supervised network to learn geometry-sensitive features for semantic matching," *In: CVPR*, 2017.

[44] S. Kim, D. Min, B. Ham, S. Lin, and K. Sohn, "Fcss: Fully convolutional self-similarity for dense semantic correspondence," *IEEE Trans. PAMI*, 2018.

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

[46] N. Ufer and B. Ommer, "Deep semantic feature matching," *In: CVPR*, 2017.

[47] K. Han, R. S. Rezende, B. Ham, K. Y. K. Wong, M. Cho, C. Schmid, and J. Ponce, "Scnet: Learning semantic correspondence," *In: ICCV*, 2017.

[48] U. Gaur and B. S. Manjunath, "Weakly supervised manifold learning of dense semantic object correspondence," *In: ICCV*, 2017.

[49] M. Tau and T. Hassner, "Dense correspondences across scenes and scales," *IEEE Trans. PAMI*, vol. 38, no. 5, pp. 875–888, 2016.

[50] I. Rocco, R. Arandjelovic, and J. Sivic, "Convolutional neural network architecture for geometric matching," *In: CVPR*, 2017.

[51] I. Rocco, R. Arandjelovic, and J. Sivic, "End-to-end weakly-supervised semantic alignment," *In:CVPR*, 2018.

[52] F. L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of image defromations," *IEEE Trans. PAMI*, vol. 11, no. 6, pp. 567–585, 1989.

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

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

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

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

[57] N. K. Bose and N. A. Ahuja, "Super-resolution and noise filtering using moving least squares," *IEEE Trans. IP*, vol. 15, no. 8, pp. 2239–2248, 2006.

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

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

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

[61] D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, and M. N. Do, "Fast global image smoothing based on weighted least squares," *TIP*, vol. 23, no. 12, pp. 5638–5653, 2014.

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

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

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

[65] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efcient alternative to sift or surf," *In: ICCV*, 2011.

[66] S. Leutenegger, M. Chli, and R. Siegwart, "Brisk : Binary robust invariant scalable keypoints," *In: ICCV*, 2011.

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

[68] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. PAMI*, vol. 35, no. 6, pp. 1397–1409, 2013.

[69] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *TPAMI*, vol. 34, no. 11, pp. 2274–2282, 2012.

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

[71] B. Ham, M. Cho, and J. Ponce, "Robust guided image filtering using nonconvex potentials," *IEEE Trans. PAMI*, 2017.

[72] I. T. Joliffe, "Principal component analysis," *Springer-Verlag*, 1986.

[73] Y. Ke and R. SukthanKar, "Pca-sift: A more distinctive representation for local image descriptors," *In: CVPR*, 2004.

[74] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *JMLR*, vol. 17, no. 1, pp. 2287–2318, 2016.

[75] M. Lang, O. Wang, T. Aydic, A. Smolic, and M. Gross, "Practical temporal consistency for image-based graphics applications," *In: SIGGRAPH*, 2012.

[76] J. Liao, Y. Yao, L. Yuan, G. Hua, and S. B. Kang, "Visual attribute transfer through deep image analogy," *In: SIGGRAPH*, 2017.

[77] J. Dong and S. Soatto, "Domain-size pooling in local descriptors: Dsp-sift," *In: CVPR*, 2015.

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

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

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

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

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

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

[84] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3d human pose annotations,," *In: ICCV*, 2009.

[85] A. Kanazawa, D. W. Jacobs, and M. Chandraker, "Warpnet: Weakly supervised matching for single-view reconstruction," *In: CVPR*, 2016.

[86] S. Kim, D. Min, B. Ham, S. Ryu, M. N. Do, and K. Sohn, "Dasc: Dense adaptive self-correlation descriptor for multi-modal and multi-spectral correspondence," *In: CVPR*, 2015.

[87] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "Matchnet: Unifying feature and metric learning for patch-based matching," *In: CVPR*, 2015.

[88] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform," *In: ECCV*, 2016.

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

Seungryong Kim received the B.S. and Ph.D. degrees in Electrical and Electronic Engineering from Yonsei University, Seoul, Korea, in 2012 and 2018, respectively. He is currently a Post-Doctoral Researcher in Electrical and Electronic Engineering at Yonsei University. His current research interests include 2D/3D computer vision, computational photography, and machine learning, in particular, sparse/dense feature descriptor and continuous/discrete optimization.

Dongbo Min received the B.S., M.S., and Ph.D. degrees from the School of Electrical and Electronic Engineering, Yonsei University, Seoul, South Korea, in 2003, 2005, and 2009, respectively. From 2009 to 2010, he was a post-doctoral researcher with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA. From 2010 to 2015, he was with the Advanced Digital Sciences Center, Singapore. From 2015 to 2018, he was an assistant professor with the Department of Computer Science and Engineering, Chungnam National University, Daejeon, South Korea. Since 2018, he has been an assistant professor with the Department of Computer Science and Engineering, Ewha Womans University, Seoul. His current research interests include computer vision, deep learning, video processing, and continuous/discrete optimization.

Stephen Lin received the B.S.E. degree in electrical engineering from Princeton University, NJ, and the Ph.D. degree in computer science and engineering from the University of Michigan, Ann Arbor. He is a Principal Researcher with the Internet Graphics Group, Microsoft Research Asia. His research interests include computer vision, image processing, and computer graphics. He served as a Program Co-Chair of the International Conference on Computer Vision 2011 and the Pacific-Rim Symposium on Image and Video Technology 2009.

Kwanghoon Sohn received the B.E. degree in electronic engineering from Yonsei University, Seoul, Korea, in 1983, the M.S.E.E. degree in electrical engineering from the University of Minnesota, Minneapolis, MN, USA, in 1985, and the Ph.D. degree in electrical and computer engineering from North Carolina State University, Raleigh, NC, USA, in 1992. He was a Senior Member of the Research engineer with the Satellite Communication Division, Electronics and Telecommunications Research Institute, Daejeon, Korea, from 1992 to 1993, and a Post-Doctoral Fellow with the MRI Center, Medical School of Georgetown University, Washington, DC, USA, in 1994. He was a Visiting Professor with Nanyang Technological University, Singapore, from 2002 to 2003. He is currently an Underwood Distinguished Professor with the School of Electrical and Electronic Engineering, Yonsei University. His research interests include 3D image processing and computer vision.