Generalized Deformable Spatial Pyramid: Geometry-Preserving Dense Correspondence Estimation

Junhwa Hur[1], Hwasup Lim[1,2], Changsoo Park[1], and Sang Chul Ahn[1,2]
[1]Center for Imaging Media Research, Robot & Media Institute, KIST, Seoul, Korea
[2]HCI & Robotics Dept., University of Science & Technology, Korea
{hurjunhwa, hslim, winspark, asc}@imrc.kist.re.kr

Abstract

We present a Generalized Deformable Spatial Pyramid (GDSP) matching algorithm for calculating the dense correspondence between a pair of images with large appearance variations. The main challenges of the problem generally originate in appearance dissimilarities and geometric variations between images. To address these challenges, we improve the existing Deformable Spatial Pyramid (DSP) [10] model by generalizing the search space and devising the spatial smoothness. The former is leveraged by rotations and scales, and the latter simultaneously considers dependencies between high-dimensional labels through the pyramid structure. Our spatial regularization in the high-dimensional space enables our model to effectively preserve the meaningful geometry of objects in the input images while allowing for a wide range of geometry variations such as perspective transform and non-rigid deformation. The experimental results on public datasets and challenging scenarios show that our method outperforms the state-of-the-art methods both qualitatively and quantitatively.

1. Introduction

Densely matching two correlated images at the pixel level is one of the most fundamental tasks in computer vision applications. A tremendous number of research efforts have been conducted in various forms. For example, narrow-baseline stereo matching finds a pixel-level correspondence along each scan-line, and 2D motion flow estimates dense correspondence fields under a small displacement constraint. In a broad concept, general dense correspondence algorithms address matching two images with different objects or scenes.

Specifically, general dense correspondence algorithms, where the object and viewpoint differ significantly, do not have explicit low-level constraints between the input images. There mainly exist two principal challenges: (1) photometric variations due to different camera settings and illumination conditions and (2) geometric variations due to viewpoint changes, object pose changes, and the non-rigid deformation of objects between the images. These various factors are projected onto the 2D space; thus, it is challenging to decompose these factors from the images.

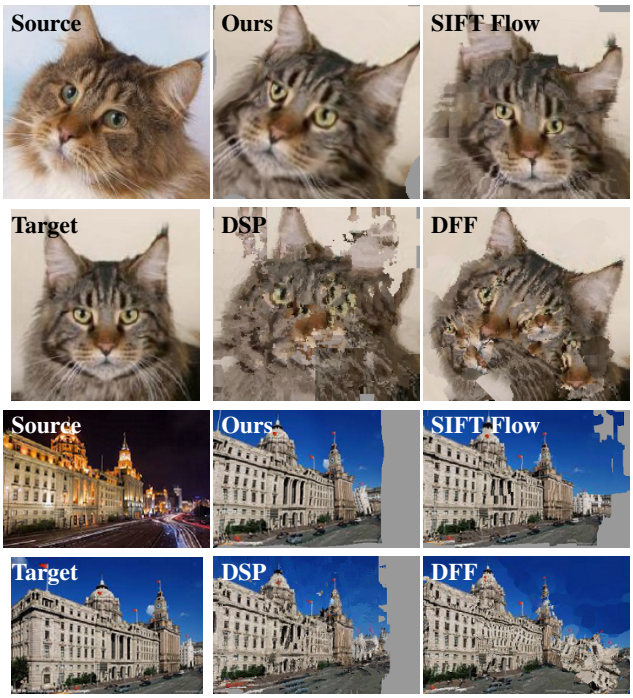


Figure 1: Backward warping results of each algorithm given a pair of images with significant geometric and photometric variations: SIFT Flow [12], DSP [10], DAISY Filter Flow (DFF) [23], and our algorithm. Images are courtesy of L. Liu, the author of [13].

Many approaches have attempted to resolve these difficulties by simplifying the complex factors. A common way to gain robustness for photometric variation is to adapt illumination-invariant descriptors, such as SIFT [14], SURF [1], or DAISY [19]. However, it is infeasible to decompose

individual factors of geometric variations. The previous attempts derive their own approximating models, such as the 2D pixel-level MRF model [11, 12, 13, 18], spatial pyramid model [10], and nearest-neighbor search [23].

However, the performance is limited to the controlled scenarios formulated by their deformation models, which failed to handle a broad range of geometric deformation. SIFT Flow [12] and Deformable Spatial Pyramid (DSP) [10] use the SIFT descriptor with a fixed scale and rotation; as a consequence, they do not fully cover the scale and rotation varying scenarios as shown in Fig. 1. To overcome the limitations on scale and rotation, DAISY Filter Flow (DFF) [23] and variants of SIFT Flow [13] increase the degree of freedom in the search space by including scale and rotation invariant properties on their descriptor. However, they show only visually plausible but geometrically incorrect matching results because they do not properly enforce spatial constraints in the spatial domain mainly due to computational concerns.

In this paper, we propose the Generalized Deformable Spatial Pyramid (GDSP) model to overcome the limitations of the previous approaches and extend the capability of matching images under versatile forms of geometric variations. We reformulate the existing DSP [10] model by imposing rotation and scale invariant properties and considering the spatial relationship in the high dimensional search space through the pyramid structure. This high dimensional regularization directly links to our main contribution: we can effectively preserve the meaningful inherent geometry and texture in images while allowing a broad range of geometric variations such as affine, perspective and even non-rigid deformation. We provide an optimization method of our high dimensional objective functions by modifying loopy belief propagation [12, 17, 22] to our formulation, which is the second contribution of our work.

The remainder of the paper is organized as follows. Section 2 reviews the previous works on dense correspondence matching and explains their weaknesses. Section 3 reviews the original DSP model and introduces our generalized model with its optimization formulation. Section 4 presents qualitative and quantitative results of our algorithms with the state-of-the-art algorithms, and Section 5 concludes the paper and suggests further work.

2. Related Work

The performance boundary of dense correspondence algorithms is generally determined by the type of descriptor and the method of regularizations that each algorithm adopts. We thoroughly examine the related research efforts on dense correspondence and describe their limitations.

SIFT Flow [12] gives an important breakthrough on dense matching between different scenes and objects. The algorithm densely extracts the SIFT descriptor for each pixel with a fixed scale and orientation, and then matches the descriptors via dual-layered loopy belief propagation. The local gradient information and pixel-level regularization enable fine matching even across different scenes or objects. However, the lack of the consideration of scale and rotation confines its scope of matching scenarios.

There have been several extensions of SIFT Flow to overcome the limitations of the original formulation. The modified algorithms employ multiple SIFT descriptors for each pixel with varying scales [18] and orientations [13] rather than fixed ones, and they match the most agreeable one among the descriptors for each pixel. The increased dimension of their descriptors expands the matching coverage of their algorithms to scale and rotation varying scenarios. Scale-Less SIFT (SLS) [8] devises a novel scale-invariant SIFT-based descriptor and demonstrates its performance in conjunction with SIFT flow.

These methods, however, overlook joint regularization between neighboring pixels in complex motion. Instead, the translation is used only for the smoothness term [8, 13], or both the translation and scale are used for the smoothness terms without considering their dependency [18]. Thus, their formulations are not sufficient to handle deformation by even a similarity transform.

DSP [10] matching suggests a coarse-to-fine approach to handle geometric deformations with orientation- and scale-fixed descriptors. The novel regularization through multiple layers in the pyramid structure efficiently handles global and local geometric deformations. A variant of DSP has also been proposed to handle rotation for 3D motion flow [9]. However, these approaches [9, 10] do not provide a concrete solution for handling severe geometric variations caused by orientation and scale changes.

More recently, DAISY Filter Flow (DFF) [23] targets non-rigid geometric variations between two images. This approach adopts the DAISY descriptor [19] and inherits the regularization scheme in the PatchMatch Filter [15], which filters the cost volume and derives optimal labels from the high dimensional search space. DFF outperforms the above previous works under scale and rotation varying scenarios. However, searching for optimal labels based on PatchMatch Filtering yields randomized results for each trial and produces visually plausible but incorrect flow fields, which are caused by weak spatial regularization.

To cope with the various types of complex deformations against the previous approaches, we suggest a generalized model of DSP [10]. We utilize orientation- and scale-varying SIFT descriptors which can effectively find correspondence when various motion exists. Additionally, we design the smoothness constraints of our model to consider the dependencies of each state through the pyramid levels, which allows it to effectively regulate and propagate the deformation of objects in a coarse-to-fine way.

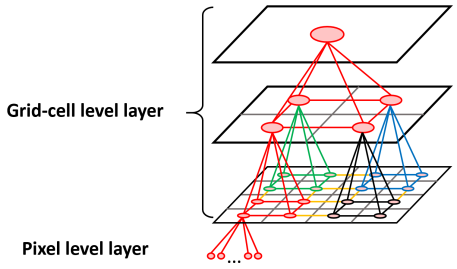


Figure 2: The graph structure of DSP model [10].

3. Approach

We first briefly review the DSP algorithm in Section 3.1 and discuss how our generalized model, GDSP, improves the shortcomings of DSP in Section 3.2. The optimization method for our objective function is described in Section 3.3.

3.1. Review on Deformable Spatial Pyramid (DSP)

Deformable Spatial Pyramid (DSP) matching [10] extracts dense correspondence across two different images in the spatial pyramid model. The pyramid model consists of two layers: a grid-cell level layer and a pixel-level layer. In the grid-cell level layer, DSP divides a source image into four rectangular cells and continues dividing each cell in the same manner until it reaches the last level of the pyramid, as shown in Fig. 2. In the graphical representation of DSP, each grid cell and each pixel correspond to nodes. The edges connect neighboring grid cells, parent-child grid cells, and each pixel with its parent grid cell.

Let i denotes an index of a node in the graphical model; $\mathbf{t}_i = (u_i, v_i)$ becomes a translation vector of node i from the source to the target image. DSP formulates its objective function in a Markov random field model [11]:

$$E(\mathbf{t}) = \sum_i D_i(\mathbf{t}_i) + \alpha \sum_{\{i,j\}\in E} V_{ij}(\mathbf{t}_i, \mathbf{t}_j). \quad (1)$$

In Eq. (1), D_i is a unary term which calculates the SIFT matching cost of node i according to its state $\mathbf{t}_i$, and V_{ij} regulates spatial smoothness between node i and node j, which are connected by an edge. DSP optimizes the objective function via loopy belief propagation.

The multiple levels of spatial extent in the DSP model allow for its robust matching on two images under large appearance variations. The upper level in the pyramid structure globally estimates matching states, while the lower level in the pyramid structure takes the responsibility of matching for local variations. However, from our experiments, DSP shows its shortcoming when handling viewpoint changes and complex object motions because the grid cells in the DSP model do not account for its scale and rotation; thus, the cells inherit incorrect estimation of its spatial moves in the final matching result.

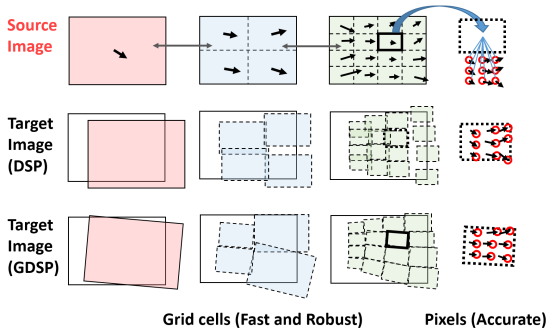


Figure 3: Comparison of matching methods between original DSP [10] model and our GDSP model.

3.2. Generalized Deformable Spatial Pyramid (GDSP)

To overcome the limitations of DSP, we propose a Generalized Deformable Spatial Pyramid (GDSP) model, which incorporates a rotation and scale term into the original model. Let I_S and I_T denote a source image and a target image to match, respectively. Our generalized objective function becomes

$$E(\mathbf{t}, r, s) = \sum_i D_i(\mathbf{t}_i, r_i, s_i) + \sum_{\{i,j\}\in E} V_{ij}(\mathbf{t}_i, r_i, s_i, \mathbf{t}_j, r_j, s_j). \quad (2)$$

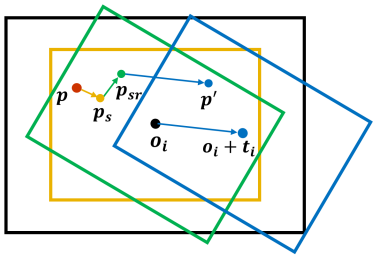
Each node i takes additional states r_i and s_i, which denote the rotation and scale in the image coordinate, respectively. Fig. 3 shows the geometric representation of our matching method compared to DSP. Our model allows each grid cell to rotate and increase or decrease itself, which gives it more flexibility to find its correspondence.

In Eq. (2), data term $D_i(\mathbf{t}_i, r_i, s_i)$ calculates the SIFT matching cost of node i given its state $(\mathbf{t}_i, r_i, s_i)$ for all sampling pixels $\mathbf{p}$ in the node:

$$D_i(\mathbf{t}_i, r_i, s_i) = \frac{1}{z} \sum_{\mathbf{p}} min(\|d_S(\mathbf{p}) - d_T(\mathbf{p}', r_i, s_i)\|_1, \lambda) \quad (3)$$

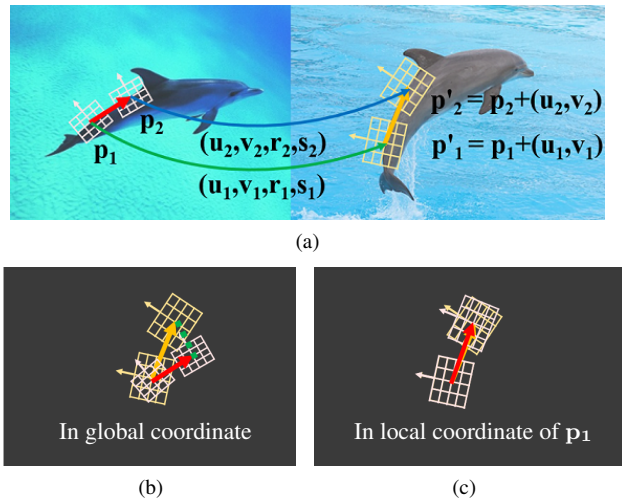
$$\mathbf{p}' = \mathbf{o}_i + s_i \cdot R(r_i) \cdot \overrightarrow{\mathbf{o}_i\mathbf{p}} + \mathbf{t}_i, \quad (4)$$

where $d_S(\mathbf{p})$ and $d_T(\mathbf{p}', r_i, s_i)$ are SIFT descriptors extracted at location $\mathbf{p}$ in the source image and at $\mathbf{p}'$ with orientation r_i and scale s_i in the target image respectively. We extract SIFT descriptors with varying r_i and s_i in the source image only, and with a fixed rotation and scale in the target image because only relative rotation and scale matter. Eq. (4) calculates a corresponding point $\mathbf{p}'$ in I_T of $\mathbf{p}$ in I_S with a given state of $(\mathbf{t}_i, r_i, s_i)$, which is visualized in Fig. 4. $R(r_i)$ is a rotation matrix which is $[\cos(r_i)$



$$\mathbf{p}_s = \mathbf{o}_i + s_i \cdot \overrightarrow{\mathbf{o}_i\mathbf{p}}$$
$$\mathbf{p}_{sr} = \mathbf{o}_i + s_i R(r_i) \cdot \overrightarrow{\mathbf{o}_i\mathbf{p}}$$
$$\mathbf{p}' = \mathbf{o}_i + s_i R(r_i) \cdot \overrightarrow{\mathbf{o}_i\mathbf{p}} + \mathbf{t}_i$$

Figure 4: Visualization of Eq. (4). **Black**: denoting a rectangular grid cell of node i centered at $\mathbf{o}_i$. **Yellow**: after adjusting scale s_i on the grid cell. **Green**: after adjusting scale s_i and rotation r_i. **Blue**: after adjusting scale s_i, rotation r_i, and translation $\mathbf{t}_i$.



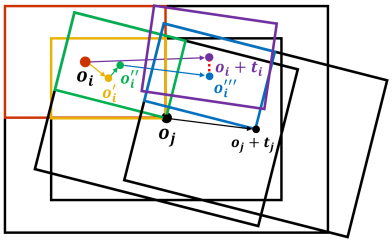
(a)

(b) (c)

Figure 5: Improper and redundant cost occurs without considering mutual dependencies for regularization (green dashed line in (b), derivation: $\|\mathbf{t}_2 - \mathbf{t}_1\|_1 = \|(\mathbf{p}'_2 - \mathbf{p}_2) - (\mathbf{p}'_1 - \mathbf{p}_1)\|_1 = \|(\mathbf{p}'_2 - \mathbf{p}'_1) - (\mathbf{p}_2 - \mathbf{p}_1)\|_1$). Almost no error occurs if mutually considering each state in the spatial domain in (c).

$-\sin(r_i)$; $\sin(r_i)\ \cos(r_i)]$, and $\mathbf{o}_i$ denotes the 2D center coordinate of node i in the source image. λ is a truncation constant, which regulates the undesired effects from outliers, and z is the number of descriptors used for calculating the data term.

The pairwise term V_{ij} in the objective function in Eq. (2) penalizes the state discrepancy of two nodes that are connected by an edge. Conventional approaches simply calculate the differences of each state individually in the parameter space and include their weighted sums in their object functions [10, 13, 18, 12]. Note that this is valid only if the regulating states are orthogonal [10, 12]. However, for the case of simultaneously regulating multiple states (scale, rotation, translation) that have dependencies [13, 18], the pairwise term should consider their mutual dependencies to



$$\mathbf{o}'_i = \mathbf{o}_j + s_j \cdot \overrightarrow{\mathbf{o}_j\mathbf{o}_i}$$
$$\mathbf{o}''_i = \mathbf{o}_j + s_j R(r_j) \cdot \overrightarrow{\mathbf{o}_j\mathbf{o}_i}$$
$$\mathbf{o}'''_i = \mathbf{o}_j + s_j R(r_j) \cdot \overrightarrow{\mathbf{o}_j\mathbf{o}_i} + \mathbf{t}_j$$

Figure 6: Derivation of V^2_{ij} in Eq. (6). **Red**: denoting a child node i of node j. **Yellow**: after adjusting scale s_j of the parent node j. **Green**: after adjusting scale s_j and rotation r_j of the parent node. **Blue**: displaying the location of node i by a state ($\mathbf{t}_j$, r_j, s_j) of the parent node j. **Purple**: indicating position of node i by its own state ($\mathbf{t}_i$, r_i, s_i). The pairwise term calculates the distance between the center points of the purple and blue rectangle (red-dashed line).

preserve the original topology of objects in images.

Fig. 5 shows an improper regularization of the previous approaches [13, 18] and our modification. When regulating states of two points $\mathbf{p}_1$, $\mathbf{p}_2$ by calculating the absolute differences of the translation states, the redundant cost occurs as the amount of the green-dashed line in Fig. 5(b). Conversely, in our modification in Fig. 5(c), we reflect the influence of rotation and scale variation on measuring the translation discrepancies by reasoning in the local spatial coordinate. This spatial reasoning provides a reasonable smoothness regularization when scale and rotation vary.

In our pyramid structure, we first impose the spatial smoothness only between nodes with the parent-child relationship, followed by the individual smoothness between nodes in the same level. This bipartite regularization propagates scale and rotation information through the pyramid structure and, at the same time, allows flexible deformation between nodes in the same level.

$$V_{ij} = \begin{cases} V^1_{ij}, \text{ if } i \text{ is a parent node} \\ V^2_{ij}, \text{ if } i \text{ is a child node} \\ V^3_{ij}, \text{ otherwise} \end{cases} \tag{5}$$

$$\begin{aligned} V^1_{ij} &= \alpha \left\| \mathbf{t}_j - ((s_i R(r_i) \cdot \overrightarrow{\mathbf{o}_i\mathbf{o}_j} - \overrightarrow{\mathbf{o}_i\mathbf{o}_j}) + \mathbf{t}_i) \right\|_1 \\ &\quad + \beta \left\| r_i - r_j \right\|_1 + \gamma \left\| s_i - s_j \right\|_1 \\ V^2_{ij} &= \alpha \left\| \mathbf{t}_i - ((s_j R(r_j) \cdot \overrightarrow{\mathbf{o}_j\mathbf{o}_i} - \overrightarrow{\mathbf{o}_j\mathbf{o}_i}) + \mathbf{t}_j) \right\|_1 \\ &\quad + \beta \left\| r_j - r_i \right\|_1 + \gamma \left\| s_i - s_j \right\|_1 \\ V^3_{ij} &= \alpha \left\| \mathbf{t}_i - \mathbf{t}_j \right\|_1 + \beta \left\| r_i - r_j \right\|_1 + \gamma \left\| s_i - s_j \right\|_1 . \end{aligned} \tag{6}$$

Eq. (6) enumerates different forms of the pairwise terms according to the relationships between node i and j (e.g.. the parent-child cells or neighbors) with weighting constants α, β, and γ. When nodes are in the parent-child relationship, we compensate the translation discrepancies to

make scale or rotation invariant when the parent cells affect the translation of their child cells. Fig. 6 displays the derivation of one case of the pairwise terms where j is a parent node and i is one of its child nodes.

3.3. Optimization

Our algorithm conducts optimization in two steps: in the grid-cell level matching and in the pixel-level matching, as in [10]. It first derives the optimal states of each grid cell by using loopy belief propagation [17, 22] and then directly calculates the optimal states for each pixel from a reduced form of the objective function.

When we optimize our objective function via loopy belief propagation, the complexity is $O(n^2)$, where n is the number of labels, which is significant in our formulation. DSP [10] employs the distance transform to compute messages for matching grid cells via loopy belief propagation to reduce the complexity to $O(n)$ [6]. Our extended pairwise terms, however, cannot directly adopt the distance transform because variables, $\mathbf{t}_i(u_i, v_i)$, r_i, and s_i are combined together in the translation-smoothness terms of V_{ij}^1 and V_{ij}^2 in Eq. (6). These terms make it difficult to directly compute translation discrepancies for passing messages.

To regard our formulation as an ordinary four dimensional case of the distance transform, we reorder the minimization term according to the dependencies of each variable. In the pairwise term V_{ij}^1 in Eq. (6), if we let $\Delta_u(r_i, s_i) = (s_i R(r_i) \cdot \overrightarrow{\mathbf{o}_i\mathbf{o}_j} - \overrightarrow{\mathbf{o}_i\mathbf{o}_j}) \cdot \hat{u}$ and $\Delta_v(r_i, s_i) = (s_i R(r_i) \cdot \overrightarrow{\mathbf{o}_i\mathbf{o}_j} - \overrightarrow{\mathbf{o}_i\mathbf{o}_j}) \cdot \hat{v}$, then a message to pass becomes:

$$
\begin{aligned}
&h(u_j, v_j, r_j, s_j) \\
&\quad = \min_{u_i, v_i, r_i, s_i} (\alpha f_1 + \alpha f_2 + \beta f_3 + \gamma f_4 + h) \\
&\quad = \min_{s_i}\{\gamma f_4 + \min_{r_i}\{\beta f_3 + \min_{v_i}[\alpha f_2 + \min_{u_i}(\alpha f_1 + h)]\}\} \\
&\qquad \text{where } f_1 = \|u_j - (\Delta_u(r_i, s_i) + u_i)\|_1 , \\
&\qquad\qquad f_2 = \|v_j - (\Delta_v(r_i, s_i) + v_i)\|_1 , \\
&\qquad\qquad f_3 = \|r_j - r_i\|_1 , f_4 = \|s_j - s_i\|_1 , \\
&\qquad\qquad \text{and } h = h(u_i, v_i, r_i, s_i).
\end{aligned}
\tag{7}
$$

The second line in Eq. (7) calculates the message that node i want to pass to node j. Because of the complex form of a minimization problem, we organize the original form into a series of four minimization problems, as in the third line of Eq. (7). Then, we can sequentially calculate the reordered form of V_{ij}^1 via a four dimensional distance transform, as shown in Algorithm 1. The algorithm updates the offset $\Delta_u(r_i, s_i)$ and $\Delta_v(r_i, s_i)$ for every r_i and s_i.

This modification successfully reduces the complexity of our optimization problem from $O(n^2)$ to $O(n)$. Additionally, the usage of the offset term suggests a general way for adopting Loopy Belief Propagation in the order of $O(n)$ when variables are linearly combined.

Algorithm 1 The Distance Transform (DT) for computing messages with V_{ij}^1 in the 4D case (u_i, v_i, r_i, s_i)

Require: A message from a parent node i to a child node j before updating, The center coordinate $\mathbf{o}_i$, $\mathbf{o}_j$ of node i and j respectively.
Ensure: A message from a parent i to a child j after updating

procedure MessageUpdating
  for s_i, r_i, v_i **do** *// calculating $h(u_j, v_i, r_i, s_i)$*
    1D DT for u_i with offset $\Delta_u(r_i, s_i)$
  end for
  for s_i, r_i, u_j **do** *// calculating $h(u_j, v_j, r_i, s_i)$*
    1D DT for v_i with offset $\Delta_v(r_i, s_i)$
  end for
  for s_i, u_j, v_j **do** *// calculating $h(u_j, v_j, r_j, s_i)$*
    1D DT for r_i
  end for
  for u_j, v_j, r_j **do** *// calculating $h(u_j, v_j, r_j, s_j)$*
    1D DT for s_i
  end for
end procedure

The pairwise term V_{ij}^2, where node i and j are a child and parent cell, respectively, follows the same procedure with the change of sign of the offset term. We give more details on the procedure in the supplementary material. The pairwise term V_{ij}^3 follows a general case of distance transform for four variables.

In the pixel-level matching, the optimal states for each pixel i can be directly derived from the objective function because no edge exists between pixels:

$$
\begin{aligned}
(\mathbf{t}_i^*, r_i^*, s_i^*) = \underset{\mathbf{t}_i, r_i, s_i}{\operatorname{argmin}} (&D_i(\mathbf{t}_i, r_i, s_i) \\
&+\alpha \left\| \mathbf{t}_i - ((s_p R(r_p) \cdot \overrightarrow{\mathbf{o}_p\mathbf{o}_i} - \overrightarrow{\mathbf{o}_p\mathbf{o}_i}) + \mathbf{t}_p) \right\|_1 \\
&+\beta \left\| r_i - r_p \right\|_1 + \gamma \left\| s_i - s_p \right\|_1 ,
\end{aligned}
\tag{8}
$$

where $\mathbf{t}_p$, r_p, s_p, and o_p denote, respectively, translation, rotation, scale, and center position of a parent node p, which were already derived in the grid-cell matching step. Finally, we use a bilateral filter for pixel-level refinement at the end of the algorithm.

4. Experiment

We verify our proposed model by comparing with the following four algorithms: SIFT Flow [12], Deformable Spatial Pyramid (DSP) [10], DAISY Filter Flow (DFF) [23], and Scale-Space SIFT Flow (SSF) [18]. We use the authors' implementations of each paper and their initial sets of parameters for fair comparisons. The experiments include quantitative and qualitative comparisons on public datasets and our own image collections. All of the algorithms were tested on a PC with Intel Core i7 3.50 GHz and 16.0GB memory.

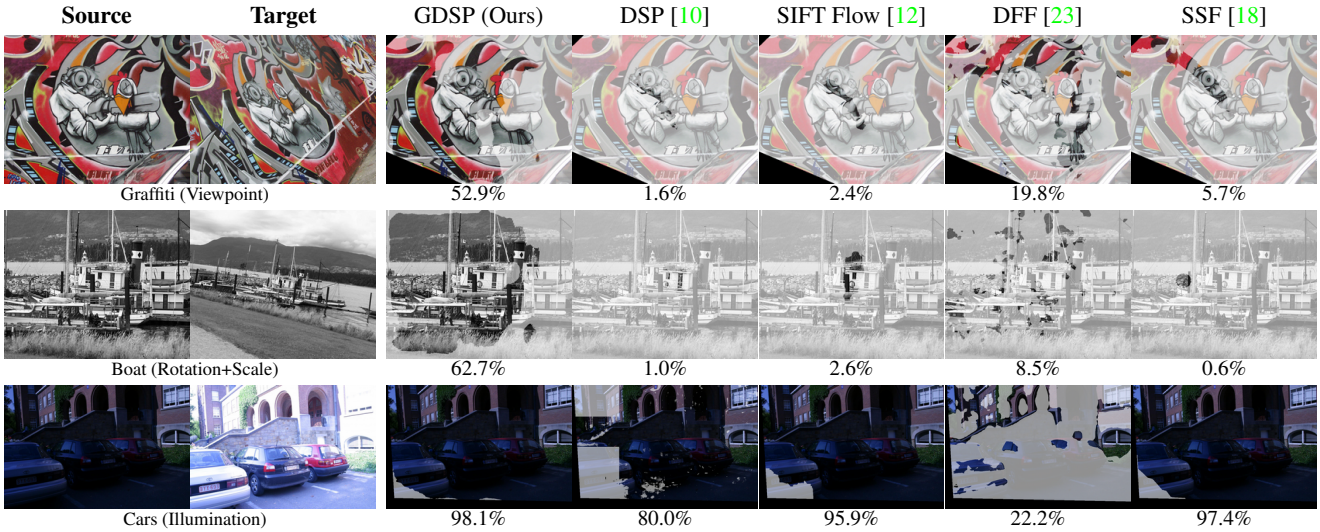


Figure 7: Several experiment results on Mikolajczyk *et al.* dataset [16] and matching percentages. On each source image, valid matching areas are colored (invalid areas, due to occlusion or dis-occlusion, are in black), and correctly matched areas are highlighted.

Scene	characteristic	**GDSP (Ours)**	DSP [10]	SIFT Flow [12]	DFF [23]	SSF [18]
Bikes	Blur	0.979 ± 0.008	0.941 ± 0.046	0.994 ± 0.005	0.766 ± 0.169	1.000 ± 0.000
Trees	Blur	0.953 ± 0.035	0.951 ± 0.036	0.946 ± 0.039	0.567 ± 0.397	0.969 ± 0.016
Graffiti	Viewpoint	0.503 ± 0.331	0.033 ± 0.029	0.238 ± 0.216	0.242 ± 0.215	0.521 ± 0.431
Bricks	Viewpoint	0.771 ± 0.417	0.230 ± 0.368	0.491 ± 0.460	0.465 ± 0.324	0.829 ± 0.282
Bark	rotation + scale	0.168 ± 0.272	0.007 ± 0.007	0.011 ± 0.017	0.018 ± 0.036	0.021 ± 0.031
Boat	rotation + scale	0.312 ± 0.208	0.003 ± 0.002	0.006 ± 0.008	0.150 ± 0.152	0.002 ± 0.001
Cars	Illumination	0.995 ± 0.008	0.858 ± 0.148	0.992 ± 0.018	0.437 ± 0.257	0.994 ± 0.012
UBC	JPEG compression	0.998 ± 0.005	0.969 ± 0.027	0.897 ± 0.068	0.753 ± 0.172	0.980 ± 0.044
Average Rank		**1.625**	**4.125**	**3.375**	**4.000**	**1.875**

Table 1: Percentages of correct match on Mikolajczyk *et al.* dataset [16]. Green colored cells and light green colored cells denote the best and the second-best statistic on each scenario, respectively.

Our algorithm extracts the dense SIFT descriptor [14] using the VLFeat library [21]. We strictly fix all parameters in our algorithm during the experiment. We empirically set the number of the pyramid level to 4, $\alpha = 0.0018$, $\beta = 0.0072$, $\gamma = 0.048$, and $\lambda = 400$. For computational efficiency, we fix the number of rotation state r_i by dividing $[-\pi, \pi]$ into 9 bins and choose 7 scale states s_i between [0.5, 2] in the log scale. For compatibility with the parameter setting, our algorithm automatically sets the width of input images to 270 pixels while keeping the width-height ratio.

Results on the Mikolajczyk *et al.* dataset [16]: We conducted an experiment on the Mikolajczyk *et al.* dataset [16] to evaluate matching performances on scene alignment. The dataset includes the same scenes taken under illumination changes, viewpoint variations, sharpness variations, planar transformation, and different compression rates.

We used the same evaluation metric as in [7, 12, 23], which calculates a percentage of correct matching pixels on a valid matching region. The computed correspondence, the error of which is less than r pixels, is considered to be correct. We relaxed the criterion and set $r = 20$ because there exist a few incorrect labels in the ground truth data that include significant geometric transformation.

Table 1 and Fig. 7 demonstrate the superior performances of our GDSP model on the dataset. Our model best estimates dense correspondence fields under illumination, compression rate, and planar scale and rotation changes. We confirm that the coverage of our model reaches to perspective transformation by referring to the results on viewpoint variation scenarios. This strength of our model comes from our high dimensional search, which includes rotation and scale variation while preserving the internal topology in images through the pyramid structure. Though our model is designed to handle significant geometric deformation, our method still demonstrates comparable results when only sharpness difference and small displacement exist.

Results on the Moseg dataset [2]: We also tested our algorithm on the Moseg Dataset [2] to evaluate how our algorithm handles large displacement and multi-layered motion. We followed the same evaluation protocols in [20, 23], which measured overlap percentages between warped images and the ground truths using the Dice coefficient [3].

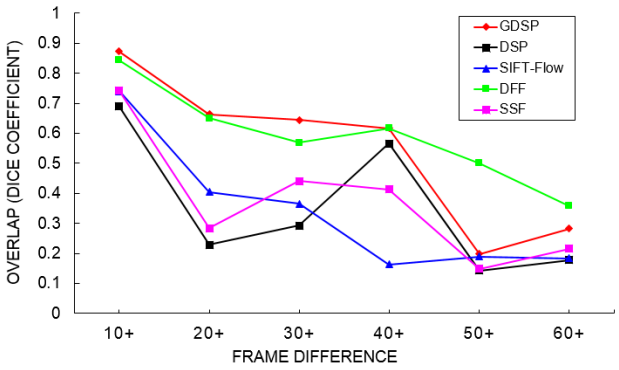


Figure 8: Quantitative results on the Moseg dataset [2].

As in Fig. 8, we confirmed that our algorithm also shows its strength on estimating dense correspondence under various multi-layered motions in outdoor scenes. The performance of our model, however, gets degraded when more than 50 frame difference occurs. This is because our pyramid-structure-based approach has its inherent difficulties on handling independent motions with substantial displacement. We expect that this issue can effectively be resolved by controlling different weights on edges in the pyramid structure via segmentation-aware approaches, which will be a part of our future work.

Results on challenging non-rigid pairs: Our geometry-preserving smoothness shows its superiority when two images specifically share similar contents and lie under non-rigid deformation. Fig. 9 shows intermediate results during our GDSP matching. An upper pyramid level coarsely holds global matching status, and, at the same time, each cell in the lower pyramid levels allow local non-rigid deformation while spatial smoothness propagates through the pyramid levels.

We tested image pairs under non-rigid deformation from our own image collections and Caltech 101 database [5]. Fig. 10 demonstrates that our model successfully computes dense correspondence fields under non-rigid deformation. The results show the backward-warped images from the targets to the sources with the obtained correspondence fields from each algorithm. The more similar the posture of warping result is to that of the source image, the more accurate the obtained dense correspondence field. DSP [10], SIFT Flow [12], and SSF [18] clearly show their inherent limitations when large scale and rotation variation exist. DFF [23] correctly delineates the shape in the source image, but its PatchMatch-based search strategy without strictly enforcing spatial coherence [15] destroys overall details with incorrect flows.

To qualitatively analyze our geometry preserving model, we track all of the corresponding pixels from the target image to the source image and generate interpolated images

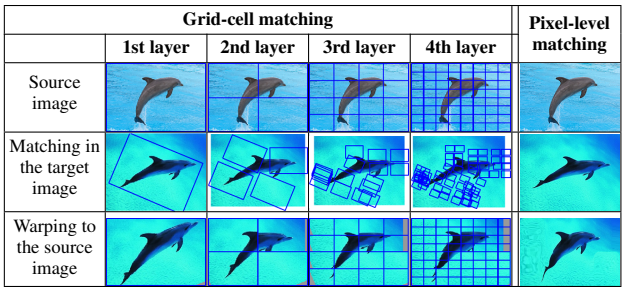


Figure 9: Intermediate results during the matching process of our model. Matching details are refined as the matching descends to the pyramid levels.

between two images. We compare our model with DFF [23], which is supposed to handle non-rigid deformation better than the other algorithms. The qualitative analysis of our model is shown in Fig. 11, and that of DFF is in the supplementary material. For each figure, the fourth column shows the backward warping results of each scenario in the two figures. The second column shows pixels in the target images, which are transferred to the warping results, and the third column demonstrates how each pixel in the target image moves to their corresponding pixel in the source image. The last column corresponds flow fields.

The interpolated sequences of our model in Fig. 11 validates that the pixel-level dense correspondence is regularly obtained while preserving the geometry of each object. The qualitative analysis of DFF's shows that the geometry of foreground objects is not preserved and the number of transferred pixels to the warping results is significantly reduced. This outcome implies that the majority of correspondences are incorrectly calculated, although the warping results look similar to their source images. The flow fields also validate these results; the flow fields of our model show piecewise-smoothness, while those of DFF [23] demonstrate irregularities overall.

Algorithm	LT-ACC	IOU	LOC-ERR
GDSP (Ours)	**0.882 ± 0.072**	**0.728 ± 0.060**	**0.078 ± 0.032**
DSP [10]	0.774 ± 0.148	0.660 ± 0.142	0.097 ± 0.041
SIFT Flow [12]	0.730 ± 0.183	0.542 ± 0.178	0.093 ± 0.028
DFF [23]	0.830 ± 0.195	0.720 ± 0.185	0.188 ± 0.086
SSF [18]	0.856 ± 0.077	0.606 ± 0.186	0.140 ± 0.053

Table 2: Quantitative analysis of non-rigid deformation scenarios using the label-transfer metrics.

Table 2 shows the quantitative analysis of the matching results as in Fig. 10. We adopt the same label-transfer metric in DSP [10]. LT-ACC denotes the accuracy of label-transfer results, which counts the percentage of correctly labelled pixels, and the intersection over union (IOU) calculates the ratio of intersection to union between the label transfer results and the ground truths [4]. LOC-ERR calculates localization errors in the bounding box. Our model outperforms the state of the arts in terms of the metrics.

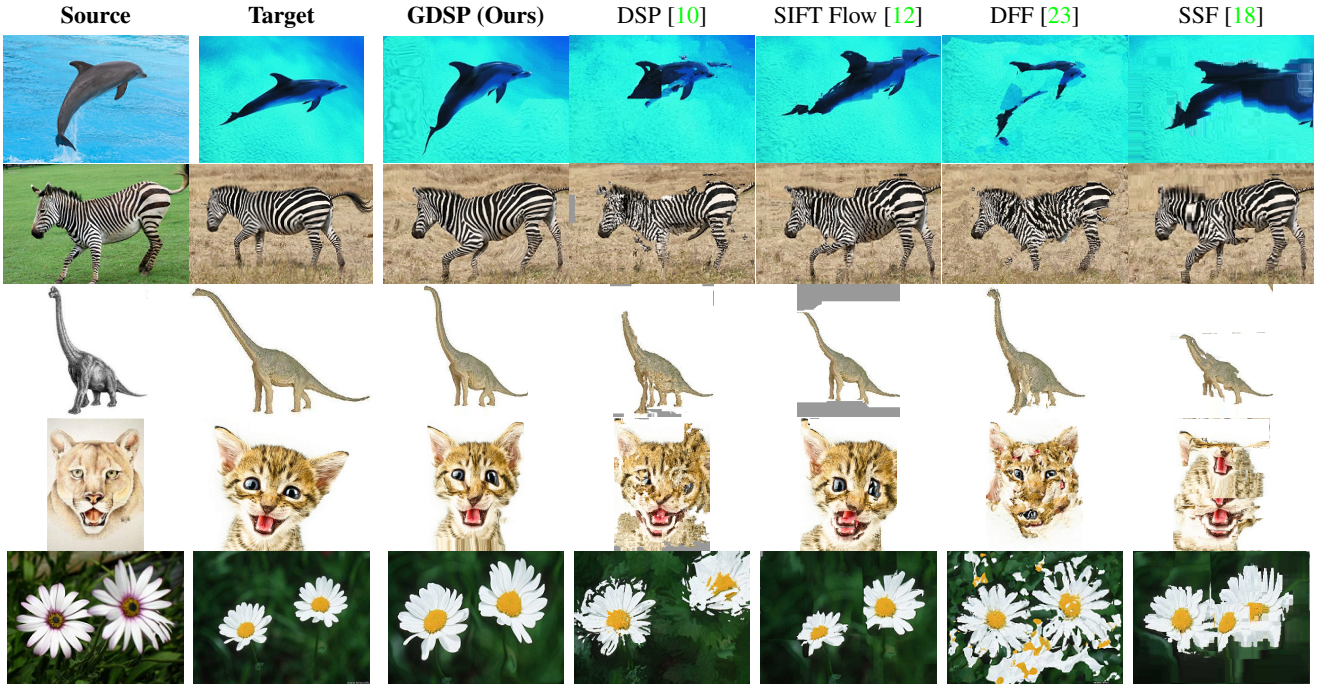


Figure 10: Backward warping results on the source images based on the obtained dense correspondence when non-rigid deformation exists. The more similar the warping result is to the source image, the more accurate the obtained dense correspondence field is.

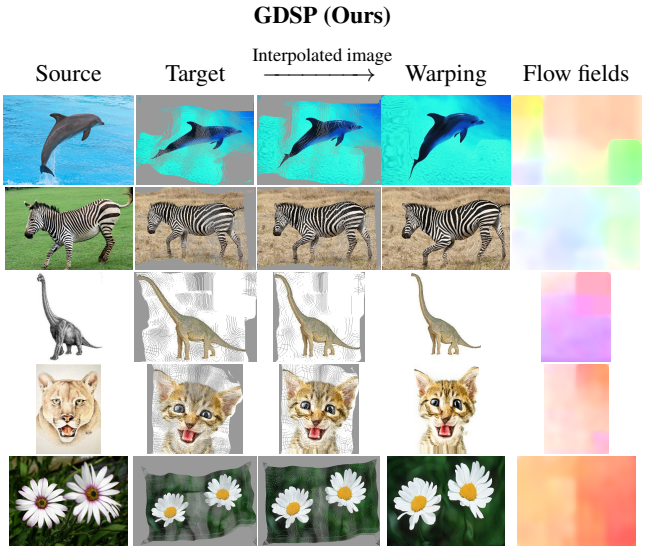


Figure 11: Qualitative analysis of our dense correspondence search results.

Runtime: The computational complexity of our algorithm inevitably becomes cubic along with the increased dimension of the search space. For handling a 320 x 240 image, on the PC with Intel Core i7 3.5 GHz and 16GB memory, our algorithm averagely takes 212.0 (s), where DSP takes 4.59 (s), SIFT-Flow 3.46 (s), DFF 64.65 (s), and SSF 37.12 (s). As in Table 3, our algorithm consumes substantial times on extracting features and matching them in the pixel level because our generalized model requires (the number of scale states) $\times$ (the number of rotation states) times more features per each sampling pixel than DSP [10]. We are, however, working on enabling parallel computing with GPU and expect the time to be substantially reduced.

Process	GDSP (Ours)	DSP [10]
Feature extraction	93.1 (s)	1.04 (s)
BP (grid-cell matching)	8.3 (s)	1.24 (s)
Pixel Matching	110.6 (s)	2.31 (s)
Total	212.0 (s)	4.59 (s)

Table 3: Runtime analysis between our algorithm and DSP [10].

5. Conclusion

We introduce a Generalized Deformable Spatial Pyramid model to extract dense correspondence between images under large photometric and geometric variations. We generalize the search space by including rotation and scale in a Loopy Belief Propagation framework. We pursue the geometry preserving smoothness in the high dimensional search space by mutually considering dependencies of each label through the pyramid structure. Our geometry preserving search successfully estimates more reliable and meaningful dense correspondence results under even non-rigid deformation compared with the state-of-the-art. Our work can be further improved by adding refinements derived from user interactions and reducing the processing time via GPU acceleration.

Acknowledgement

We would like to thank Dr. Young Min Kim for many helpful comments on this paper, and we also truly thank our anonymous reviewers for their in-depth reviews.

This work was supported by the Global Frontier R&D Program on <Human-centered Interaction for Coexistence> funded by the National Research Foundation of Korea grant funded by the Korean Government(MSIP) (2010-0029752).

References

[1] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *CVIU*, 110(3), 2008. 1

[2] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010. 6, 7

[3] L. R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26, 1945. 6

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88, 2010. 7

[5] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPRW*, 2004. 7

[6] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70(1), 2006. 5

[7] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski. Non-rigid dense correspondence with applications for image enhancement. *ACM TOG*, 30(4), 2011. 6

[8] T. Hassner, V. Mayzels, and L. Zelnik-Manor. On sifts and their scales. In *CVPR*, 2012. 2

[9] J. Hur, H. Lim, and S. C. Ahn. 3D deformable spatial pyramid for dense 3d motion flow of deformable object. In *ISVC*, 2014. 2

[10] J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *CVPR*, 2013. 1, 2, 3, 4, 5, 6, 7, 8

[11] S. Z. Li and S. Singh. *Markov random field modeling in image analysis*, volume 26. Springer, 2009. 2, 3

[12] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Trans. PAMI*, 33(5), 2011. 1, 2, 4, 5, 6, 7, 8

[13] L. Liu, K.-L. Low, and W.-Y. Lin. Dense image correspondence under large appearance variations. In *ICIP*, 2013. 1, 2, 4

[14] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2), 2004. 1, 6

[15] J. Lu, H. Yang, D. Min, and M. Do. Patch match filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation. In *CVPR*, 2013. 2, 7

[16] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *IJCV*, 65(1-2), 2005. 6

[17] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *In UAI*, 1999. 2, 5

[18] W. Qiu, X. Wang, X. Bai, A. Yuille, and Z. Tu. Scale-space sift flow. In *WACV*, 2014. 2, 4, 5, 6, 7, 8

[19] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide baseline stereo. *IEEE Trans. PAMI*, 32(5), 2010. 1, 2

[20] E. Trulls, I. Kokkinos, A. Sanfeliu, and F. Moreno-Noguer. Dense segmentation-aware descriptors. *CVPR*, 2013. 6

[21] A. Vedaldi and B. Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008. 6

[22] Y. Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Comput.*, 12(1), 2000. 2, 5

[23] H. Yang, W.-Y. Lin, and J. Lu. Daisy filter flow: A generalized discrete approach to dense correspondences. In *CVPR*, 2014. 1, 2, 5, 6, 7, 8