DNN Flow: DNN Feature Pyramid based Image Matching

Wei Yu*[1]
w.yu@hit.edu.cn
Kuiyuan Yang[2]
kuyang@microsoft.com
Yalong Bai[1]
ylbai@mtlab.hit.edu.cn
Hongxun Yao[1]
h.yao@hit.edu.cn
Yong Rui[2]
yongrui@microsoft.com

[1] Harbin Institute of Technology
Harbin, China
[2] Microsoft Research
Beijing, China

Abstract

Image matching especially in category level is a challenge but important problem in vision. The advance of image matching largely depends on the advance of image features. In viewing recent success of learned image feature by DNN, we propose an image matching algorithm based on DNN feature pyramid, named as DNN Flow. The nature of DNN feature pyramid in detecting different level patterns makes it is suitable to match two images in a coarse to fine manner, where top level coarsely matches two images in object level, middle level matches two images in part level, and low level finely matches two images in pixel level. The coarse to fine matching based on DNN feature pyramid is formulated as a series of optimization problems considering the guidance from top level. Extensive experiments demonstrate the superiority of DNN Flow in image matching under challenge variations.

1 Introduction

Image matching is a fundamental problem in computer vision, which is the cornerstone for many vision problems, such as motion estimation [11] [2], label propagation [14] [5] and object modeling [7] [1] [10]. The goal of image matching is to find the corresponding pixels between two images. Based on the variations between the two images, we roughly divide image matching into two categories, i.e., instance-level matching [15] and category-level matching [16]. In instance-level matching, the two images can be of the same object varied by motion, or the same scene with affine transformations. While in category-level matching, the two images are about objects/scenes of same category with more challenge variations. These variations arise not only from changes in illumination and viewpoint, but also appearance variations due to different object instances. Category-level matching aims

*This work was done when Wei Yu was an intern at Microsoft Research.

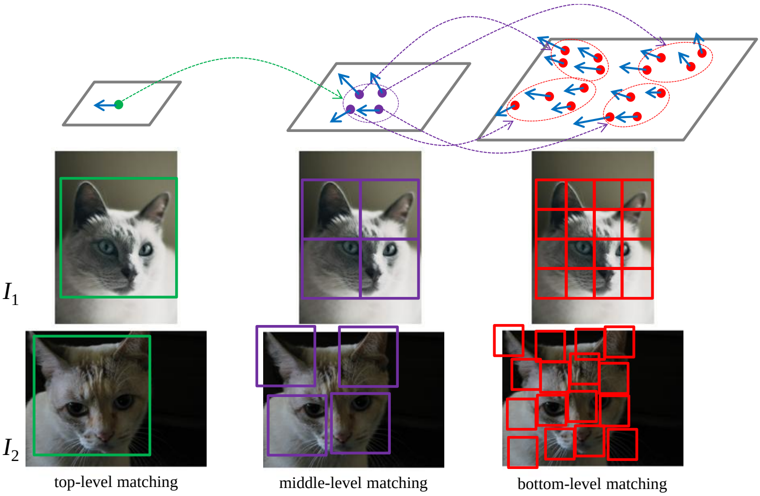


Figure 1: Matching I_1 and I_2 using DNN flow. Each column shows the matching of different levels. In first row, parallelogram denotes the DNN feature image of I_1, where dot represents the feature at that location. Line with arrow denotes the flow vector of the corresponding feature, while curve with arrow denotes guidance from high level to low level. In second row, the color rectangles show I_1's patches covered by DNN features in first row. Third row shows I_2's matching patches corresponding to the patches in second row.

to overcome the intra-class variability in shape and other visual properties, such as cars with various shapes and colors and cats with different poses and furs.

Image matching is achieved through finding similar locations in the two images. Here, the similarity between locations is typically measured by appearance and geometric constraint. In general, appearance is determined by image features extracted at each location, while geometric constraint is preselected such as smoothness and small displacements [11]. The image features for image matching are required with different invariance abilities for different image matching problems. For example, intensity is directly used as image features to match two adjacent video frames, where only exists small variations caused by motion [6]. To match two images with affine transformations, SIFT [13] is used as the image feature and has achieved great success in such case [12]. Generally, current image features are with well invariance ability for instance-level matching. Unfortunately, these features still cannot handle the intra-class variations for category-level matching.

Recently, Deep Neural Network (DNN) has achieved state-of-the-art performance in image classification [9], and showed great ability in handling the variations under the same category. The ability comes from the gradual abstraction through several levels, where low level detects simple patterns, such as edges and blobs, middle level detects object parts and high level detects objects.

Considering the ability of DNN feature in handling semantic variations, we propose a novel image matching method based on DNN feature pyramid, named as DNN Flow. DNN Flow utilizes DNN features of different levels to achieve coarse to fine matching. As shown in Figure 1, top level matching attempts to achieve object level matching since top level features detect patterns at object level, middle level matching establishes correspondences at part level, finally bottom level matching achieves fine level matching through small patterns.

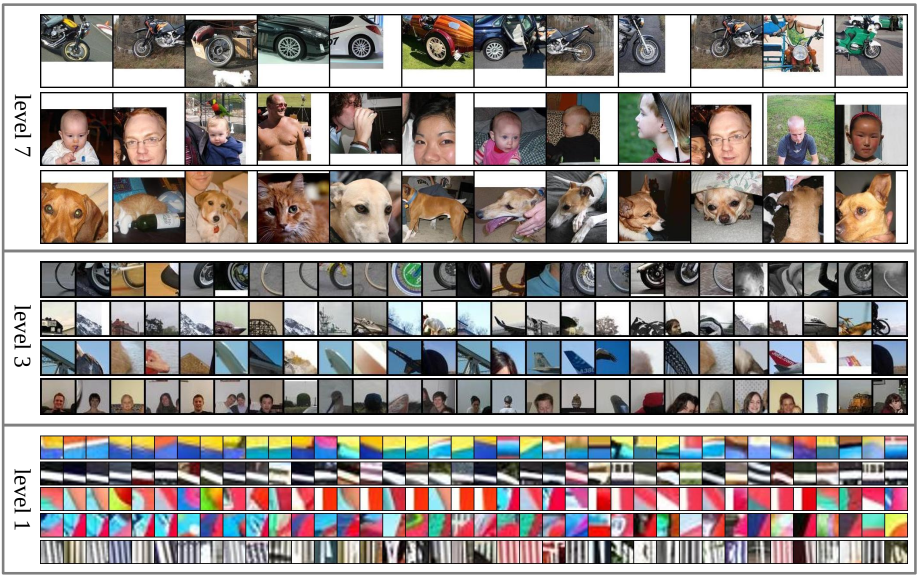


Figure 2: The sample patches corresponding to top activations on some dimensions of DNN features from different levels. Each row shows the patches which with largest value in the corresponding dimension. The hierarchic details of DNN will be illustrated in following sections, where level 7 is top level and level 1 is bottom level. The low level features describe simple patterns, such as edge and stripe. The high level features describe object-level patterns, while the middle-level feature describe part-level patterns.

The main advantage of DNN flow is to utilize more targeted feature to achieve the matching goal of each level. The top level feature with semantic invariance helps to discriminate inter-class variance and stand intra-class variance. Therefore, top level feature is robust to fight against various visual variance. Even if two images of the same category are obvious similar at bottom level, high-level matching still produces helpful coarse flow field and guides low-level matching along with the reasonable direction.

Figure 2 illustrates some patches corresponding to the top activations on some filters learned by DNN. On one hand, the dimensions of top level feature response same object class, which make top level features with less ambiguity and good distinction. On the other hand, the dimensions of bottom level feature response the patches with similar simple patterns and with more ambiguity. The dimensions of other middle level feature response part, which are appropriate as mediate level matching.

In order to demonstrate the effectiveness of DNNFlow, we select three state-of-the-art approaches as baselines, PatchMatch [1], SIFT Flow[12] and DSP [8]. These approaches all focus on estimating dense correspondences through a coarse-to-fine process and achieve competitive performance. We will compare against these three approaches on three datasets for three tasks respectively: rough image dense matching, fine object alignment and label transfer.

2 Related work

As previously mentioned, current approaches attempt to match images to estimate dense correspondences with more challenge variations. The challenge introduces a problem of efficiency that spatial searching range spreads even from neighbors to whole image. In order to avoid massive computational cost, recent approaches has to consider how to balance efficiency and effectiveness, and the coarse-to-fine strategy is a feasible way to produce receivable correspondences matching .

Patch Match [1] estimates dense correspondences based on randomized search technique. It proposes a extend matching strategy that provides *k* nearest neighbors instead of only one to search across scales and rotations. Patch Match algorithm achieves a implicity coarse-to-fine strategy. Some reliable matchings are estimated firstly, then the reliable matchings will guide nearby locations' matching.

SIFT Flow [11] improves the flow field between two images by using SIFT features, and greatly speeds up the matching process by using coase-to-fine strategy. However, the coase level feature is with limited information as it is a downsampled version of original SIFT image.

DSP [8] also builds a hierarchical matching framework to estimate dense correspondence. Through regularizing matching consistency at different levels, DSP estimates correspondences from whole image, to coarse grid cells, to each pixel. Considering the information loss in coarse level, DSP tries to preserve richer visual information using multiple descriptors on the bottom level. DSP starts from the entire image matching, then keeps dividing upper-level image into four lower-level grid cells and estimate s correspondences on each level.

In a hierarchical framework, coase-level features need discriminate inter-class variance and stand intra-class variance. However, local gradient based features are inevitable to lost useful visual information in the process of down-sampling. With a intuitive view, coarse matching on top level aims to search the region with same label, while fine matching on bottom level focuses on aligning the pixels with similar visual details. Once the correspondences of regions are estimated exactly, the inner pixels can be correctly matched with greater probability. Therefore, DNN features are appropriate to match image based on coarse-to-fine strategy.

The essential difference of the proposed DNN flow with the above three methods is that it use DNN feature pyramid to achieve coarse to fine matching, where coarse level matching is via object level patterns in contrast to low level descriptors used in other methods.

3 DNN Flow Approach

In this section, we firstly introduce DNN feature used in DNN flow. Then, we present the process of coarse-to-fine matching in our DNN Flow framework. Finally, we introduce the matching objective in the hierarchical framework.

3.1 DNN Feature

The DNN used for extracting DNN feature is learned by supervised back propagation on ILSVRC2012 training set, which is similar to [9]. As shown in Figure 3, the used DNN contains eight layers with weights: five convolutional layers followed by three fully-connected

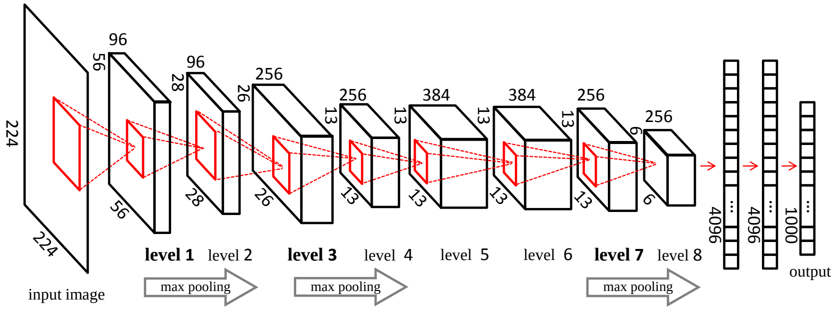


Figure 3: Schematic illustration of the used DNN structure. The DNN features are extracted from left to right, where level 1, 3, 5, 6, 7 are the outputs of convolutional layers and level 2, 4, 8 are the outputs of pooling layers.

layers. Three max-pooling layers are used following the first, seconde and fifth convolutional levels. Considering the following three reasons: (1) the outputs of fully-connected layers are corresponding the whole image and contain no location information for matching, (2) max-pooling layer detects the same patterns as its former convolutional level, (3) the outputs of the level 5 and level 6 are of same size with level 7. Thus we only use the outputs of level 1, level 3 and level 7 to build the DNN feature pyramid.

3.2 Framework of DNN Flow

As mentioned in Sec. 3.1, we adopt the output of level 7 as top-level feature and the output of level 3 as mid-level feature. Here, the size of input image is 224×224 while the size of our bottom-level feature is 56×56, since the nature of DNN. Thus, we adjust step of first convolutional level to produce a dense feature mapping with size 224×224. We adopt the dense feature mapping as bottom-level feature and original output of level 1 as another mid-level feature.

DNN Flow builds a four-level pyramid to estimate dense correspondences. Considering the overlap between neighbor filters in specific DNN levels, each flow vector produced by upper-level matching guides four lower-level features matching, until reaching the bottom level. Top-level matching estimates a coarse flow field, which presents correspondences between the regions with similar semantic. Iteratively, the upper-level flow filed guides lower-level flow field optimizing.

3.3 Matching Objective Function

Given two images I_1, I_2, and their DNN feature pyramids F_1 and F_2. Let $p = (x,y)$ be the grid coordinate in the feature pyramid, and $F_1(p,i)$ denotes the feature at p on the i^{th} level. Let w_i be the flow field on the i^{th} level, and $w_i(p) = (u_i(p), v_i(p))$ be the flow vector at p, where $u_i(p)$ and $v_i(p)$ are horizontal flow vector and vertical flow vector respectively. Then, the DNN Flow's matching objective function is formulated as:

$$E(w_i|w_{i-1},i) = \sum_p (E_D(p,w_i) + \alpha \sum_{q\in\varepsilon(p,i)} E_S(p,q,w_i) + \beta E_{SD}(p,w_i,w_{i-1})), i \in \{1,2,3\} \quad (1)$$

where E_D, E_S, E_{SD} are the data term, smoothness term and small displacement term respectively, $\varepsilon(p,i)$ is the neighborhoods of p on the i^{th} level. The detailed formulation of the three terms E_D, E_S, E_{SD} are defined as following:

$$E_D(p,w_i) = |F_1(p,i) - F_2(p + w_i(p),i)| \tag{2}$$

$$E_S(p,q,w_i) = |u_i(p) - u_i(q)| + |v_i(p) - v_i(q)| \tag{3}$$

$$E_{SD}(p,w_i,w_{i-1}) = |u_i(p) - \widetilde{u}_{i-1}(p)| + |v_i(p) - \widetilde{v}_{i-1}(p)| \tag{4}$$

where $(\widetilde{u}_{i-1}, \widetilde{v}_{i-1})$ is the w_{i-1} mapped to i^{th} level based on mapping of DNN.

Data term E_D measures the similarity between the corresponding features on the same level. Smoothness term E_S leverages the geometric prior that neighbors' flow vectors should be similar. Small displacement term E_{SD} uses the flow field of upper level to guide the optimization of low-level flow field. In particular, w_0 is initialized as zero.

4 Experiment

In this section, we verify the DNN flow through three experiments: rough image dense matching, fine object alignment and label transfer. Three state-of-the-art methods including DSP, SIFT Flow and PatchMatch which based on local feature or hierarchical local feature are compared in all experiments.

4.1 Evaluation Metrics

In order to quantitatively evaluate image matching, two evaluation metrics are introduced into experiment: label transfer accuracy (**LT-ACC**) metric [12] and intersection over union (**IOU**) metric [3].

For two images I_1 and I_2 annotated with pixel-level labels, the pixel-level labels of one image are transferred to another image along with the estimated flow field. Let L_1 and L_2 be the label maps of I_1 and I_2 respectively, L'_1 be the label map warped from L_2 to L_1 along with flow field w, where $L'_1(p) = L_2(p + w(p))$ for $\forall p \in I_1$.

LT-ACC metric measures the proportion of pixels labeled correctly, and defined as:

$$\text{LT-ACC} = \frac{area(L_1 \bigcap L'_1)}{area(L_1)} \tag{5}$$

IOU metric mainly measures the matching quality of foreground object, which assumes only one foreground object in an image. That means the labels of images are binary, one denotes foreground region and the other denotes background region. And IOU is defined as:

$$\text{IOU} = \frac{area((L_1 = 1) \bigcap (L'_1 = 1))}{area((L_1 = 1) \bigcup (L'_1 = 1))} \tag{6}$$

Table 1: Rough image dense matching on Caltech101.

	DNNFlow	DSP	SIFT Flow	PatchMatch
LT-ACC	**0.775**	0.732	0.684	0.646
IOU	**0.493**	0.482	0.450	0.375

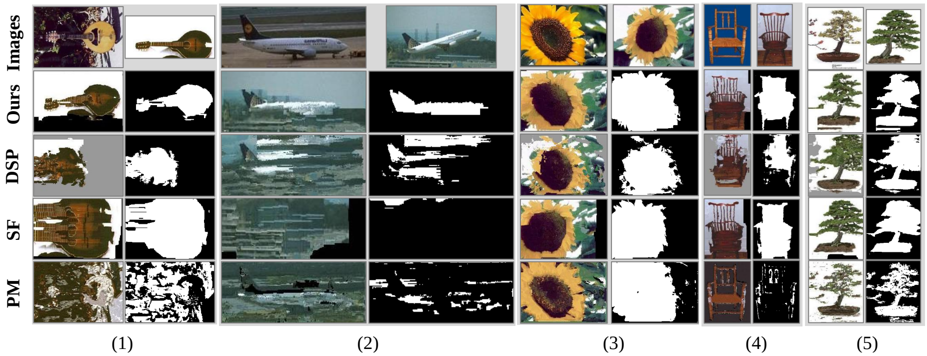


Figure 4: Examples of image matched by different approaches (SF means SIFT Flow and PM means Patch Match). The first row shows five pairs of images to match, while row 2-5 show the warping result based on the correspondence estimated by different approaches. In row 2-5, left column shows the warping image from second image to first image, and right column shows the transferred label where white region is foreground and black region is background).

4.2 Rough Image Dense Matching

Firstly, we evaluate rough image dense matching on Caltech101 [4], which aims to match images of the same category. Caltech101 contains 101 object classes, where each image is annotated by pixel-level labels. For fair comparison, we follow the same experiment setting as DSP [8], where 15 image pairs are randomly selected from each category in Caltech101.

The quantitative evaluation results are shown in Table 1. DNN flow can outperform D-SP by 4 points in LT-ACC, while outperform SIFT flow by 9 points. Figure 4 shows the matching results by different approaches. DNN flow can work better under different object variations, such as cluttering background (1st example), diverse pose (2nd example), different viewpoints (3rd example), disparate intra-class appearance (4th example) and various scales (5th example).

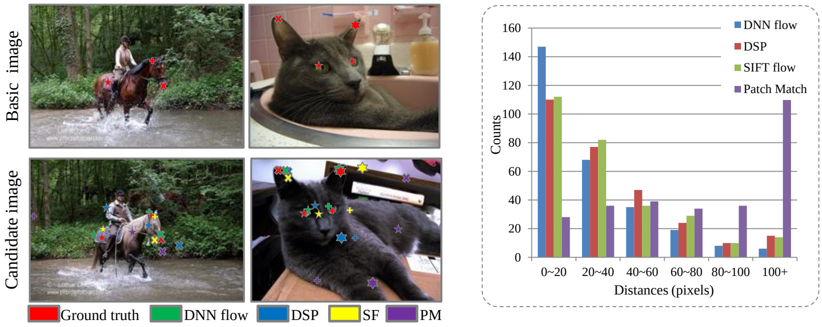
4.3 Fine Object Alignment

In order to further illustrate the validity of DNN Flow, we conduct an object alignment experiment on a more challenge dataset Pascal VOC2007, which focuses on matching key points between objects from the same category in two different images. We randomly select 100 images from Pascal VOC2007 as basic images , and find their nearest neighbors in GIST feature space as candidate images to match. For each basic image and its candidate image, several key points with strong semantic meanings are manually annotated as illustrated in Fig. 5(a). The matching results of different methods are also showed in Fig. 5(a).

We quantitatively compare DNN Flow with other baselines using average L1 distance between coordinates of annotated key points and their corresponding warped key points. The

Table 2: Object matching on Pascal VOC2007.

	DNNFlow	DSP	SIFT Flow	PatchMatch
Average distance	**26.2**	33.0	34.5	91.9



(a) Examples of fine object alignment

(b) Histogram of average distance

Figure 5: Examples of fine object alignment and the histogram of distance. In (a), different key points in basic images are marked by different shapes. In candidate image, the ground truth key points and the matched points estimated by different approaches are marked by different colors.(b) shows the statistics of the distance between key points from ground truth and estimated by different matching methods.

results are summarized in Table 2, where DNN Flow achieved the smallest average distance and demonstrates the effectiveness of DNN flow in finding the detailed correspondence with challenge variations.

4.4 Label Transfer

In this section, we validate DNN Flow through scene matching, while the previous two experiments both focus on matching objects. In the experiment of label transfer, a test image with no label information firstly searches the most similar image from the candidate set where the candidate images have been labeled. Then, the labels can be transferred from the selected candidate image to the test image. In this experiment, we also follow the same experiment setting with DSP [8] on LMO dataset. LMO contains 2688 images with pixel-level labels from 33 classes, details of the labels are shown in Figure 6. We randomly select half number of images as unlabeled images, and find their nearest neighbors as candidate images to transfer labels.

LT-ACC is used to measure the label transfer accuracy since IOU only suitable for images with only one foreground object. The quantitative results are summarized in Table 3. As suggested in [12], the matchable pixels that don't belong to the common classes of two images are ignored. Figure 6 shows some examples of label transfer. Although DSP and SIFT Flow output reasonable label transfer, DNN Flow can better handle intra-class variations and achieved much higher label transfer accuracy. Several label transfer examples are showed in Figure 6.

Table 3: Scene matching on LMO.

	DNNFlow	DSP	SIFT Flow	PatchMatch
LT-ACC	**0.737**	0.706	0.672	0.607

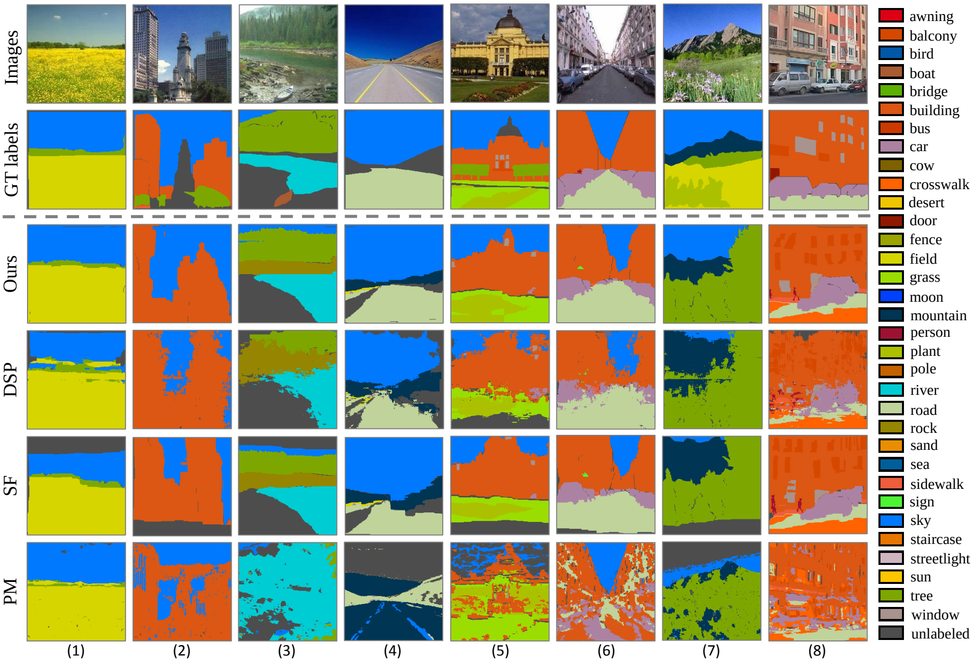


Figure 6: Examples of label transfer by different approaches. 7th column and 8th column show our failure examples, where 7th example lacks of common classes and 8th example fail to match single car to multiple cars.

5 Conclusion

We propose a DNN feature pyramid based hierarchical image matching framework to estimate dense correspondences between two category-level scenes. Compared with the tradition approaches, DNN Flow matches two images in a coarse to fine manner via complex to simple pattern detectors in DNN model. Through extensive experiments on different datasets, DNN Flow has shown its capability in category-level matching under more challenge variations.

References

[1] Connelly Barnes, Eli Shechtman, Dan B Goldman, and Adam Finkelstein. The generalized patchmatch correspondence algorithm. In *ECCV*. 2010.

[2] Thomas Brox and Jitendra Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *PAMI*, 33(3):500–513, 2011.

[3] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew

Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

[4] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *CVIU*, 106(1):59–70, 2007.

[5] Stephen Gould and Yuhang Zhang. Patchmatchgraph: building a graph of dense patch correspondences for label transfer. In *ECCV*. 2012.

[6] Berthold K Horn and Brian G Schunck. Determining optical flow. In *1981 Technical Symposium East*, pages 319–331, 1981.

[7] Hao Jiang and SX Yu. Linear solution to scale and rotation invariant object matching. In *CVPR*, 2009.

[8] Jaechul Kim, Ce Liu, Fei Sha, and Kristen Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *CVPR*, 2013.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, volume 1, page 4, 2012.

[10] Yan Li, Leon Gu, and Takeo Kanade. Robustly aligning a shape model and its application to car alignment of unknown pose. *PAMI*, 33(9):1860–1876, 2011.

[11] Ce Liu, Jenny Yuen, Antonio Torralba, Josef Sivic, and William T Freeman. Sift flow: Dense correspondence across different scenes. In *ECCV*. 2008.

[12] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *PAMI*, 33(5):978–994, 2011.

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

[14] Michael Rubinstein, Ce Liu, and William T Freeman. Annotation propagation in large image databases via dense image correspondence. In *ECCV*. 2012.

[15] Pengfei Xu, Lei Zhang, Kuiyuan Yang, and Hongxun Yao. Nested-sift for efficient image matching and retrieval. In *IEEE MultiMedia*. 2013.

[16] Wei Yu, Hongxun Yao, Kuiyuan Yang, and Lei Zhang. The shortest warping path based multiple images alignment. In *ICIP*. 2013.