Joint Recovery of Dense Correspondence and Cosegmentation in Two Images — Supplementary Material —

Tatsunori Taniai
The University of Tokyo

Sudipta N. Sinha
Microsoft Research

Yoichi Sato
The University of Tokyo

In the supplementary material we present derivations and proofs associated with the proposed technique that were omitted from the main paper due to lack of space. Some additional notes and implementation details are also provided. We will be referring to certain equations and figures in the main paper. Please note that the new equations and figures provided in the supplementary material have numbers with the letter A as prefix to distinguish them from those in the main paper. We also provide additional qualitative comparisons as a supplementary video on our project website.

A. Continuous Alpha Map Formulation

Here, we explain why in our method, the per-pixel segmentation labels must be continuous alpha-matte values $\alpha \in [0, 1]$ rather than binary values $\{0, 1\}$. If α were binary, the flows $\mathbf{T}_i$ at nodes labeled background ($\alpha_i = 0$) would be under-constrained, because the flow data term $\mathcal{E}^i_{\text{flo}}$ in Eq. (3) at such nodes would always be a constant λ_{occ} regardless of the values of $\mathbf{T}_i$. This would be problematic, because if true foreground nodes are incorrectly labeled background in early stages of our inference process, it would be harder to recover their true flow labels in later iterations. To avoid this issue, we require α to be a continuous value that is larger than a small positive value (0.1 in our implementation). By doing this we will have meaningful flow labels $\mathbf{T}_i$ even at nodes labeled (incorrectly) background, because those flow labels still slightly affect matching energies of $\mathcal{E}^i_{\text{flo}}$.

B. Energy Approximation

Next, we present the derivations of our approximation energy described in Section 4.1.

We first derive the energy function $\mathcal{E}(f, G^{k+1})$ in the form of Eq. (15). In order to simplify the energy formulation in Eq. (8), we denote energies involved in each layer as

$$\mathcal{E}^l_{\text{lay}}(f, G) = \mathcal{E}_{\text{mrf}}(f|L_l) + \mathcal{E}^l_{\text{reg}}(f|G) + \mathcal{E}^l_{\text{gra}}(V_l) \tag{A1}$$

and rewrite the energy function $\mathcal{E}(f, G^{k+1})$ as the sum of layer energies

$$\mathcal{E}(f, G^{k+1}) = \sum_{l=0}^{k+1} \mathcal{E}^l_{\text{lay}}(f, G^{k+1}) \tag{A2}$$

$$= \sum_{l=0}^{k} \mathcal{E}^l_{\text{lay}}(f, G^k) + \mathcal{E}^{k+1}_{\text{lay}}(f, G^{k+1}) \tag{A3}$$

$$= \underbrace{\mathcal{E}(f, G^k)}_{\mathcal{E}(f|G^k)} + \underbrace{\mathcal{E}^{k+1}_{\text{lay}}(f, G^{k+1})}_{\mathcal{E}_{\text{top}}(f, L^{k+1})}. \tag{A4}$$

Assuming that G^k is known from the previous iteration, we denote $\mathcal{E}(f, G^k)$ as $\mathcal{E}(f|G^k)$, and $\mathcal{E}^{k+1}_{\text{lay}}(f, G^{k+1})$ as $\mathcal{E}_{\text{top}}(f, L^{k+1})$

to obtain Eq. (15).

To approximate the above $\mathcal{E}(f, G^{k+1})$, we create a temporary graph $\hat{G}^{k+1}$ as an approximation of G^{k+1}, by duplicating the top layer of G^k as $L'_k = (V'_k, E'_k) \leftarrow (V_k, E_k)$. We further define a labeling $\hat{f}$ on this temporary graph $\hat{G}^{k+1}$. Since f and $\hat{f}$ are defined on the different graphs (G^{k+1} and $\hat{G}^{k+1}$) or different top layers (V_{k+1} and V'_k), we cannot simply assume $f = \hat{f}$. However, V'_k representing a superpixel segmentation is the finest form of any possible V_{k+1} due to the tree structure of G. Therefore, we can always define $\hat{f}$ so that f and $\hat{f}$ are equivalent $f \equiv \hat{f}$, *i.e.*, the pixelwise labeling included by both f and $\hat{f}$ are identical.

Using $\hat{f}$ and $\hat{G}^{k+1}$, our approximation function $\hat{\mathcal{E}}(\hat{f}|\hat{G}^{k+1})$ for $\mathcal{E}(f, G^{k+1})$ in the form of Eq. (17) is obtained by substituting $G^{k+1} \leftarrow \hat{G}^{k+1}$ and $f \leftarrow \hat{f}$ into $\mathcal{E}(f, G^{k+1})$.

$$\hat{\mathcal{E}}(\hat{f}|\hat{G}^{k+1}) = \mathcal{E}(\hat{f}, \hat{G}^{k+1}) \tag{A5}$$

$$= \underbrace{\mathcal{E}(\hat{f}, G^k)}_{\mathcal{E}(\hat{f}|G^k)} + \underbrace{\mathcal{E}^{k+1}_{\text{lay}}(\hat{f}, \hat{G}^{k+1})}_{\mathcal{A}(\hat{f})}. \tag{A6}$$

Here, because G^{k+1} and $\hat{G}^{k+1}$ share the same structure except for the top layers, the energies $\mathcal{E}(\cdot|G^k)$ involved in the bottom hierarchy G^k are equivalent between Eqs. (A4) and (A6). To discuss how $\mathcal{A}(\hat{f})$ approximates $\mathcal{E}^{k+1}_{\text{top}}(f, L^{k+1})$, we write it as

$$\mathcal{A}(\hat{f}) = \lambda_{\text{flo}} \sum_{i \in V'_k} \mathcal{E}^i_{\text{flo}}(\hat{f}_i) + \lambda_{\text{seg}} \sum_{i \in V'_k} \mathcal{E}^i_{\text{seg}}(\hat{f}_i) + \sum_{(s,t) \in E'_k} w_{st}\, \mathcal{E}^{st}_{\text{reg}}(\hat{f}_s, \hat{f}_t) + \sum_{(p,c) \in E^{\text{pc}'}_{k+1}} w_{pc}\, \mathcal{E}^{pc}_{\text{reg}}(f_p, f_c) + \mathcal{E}^{k+1}_{\text{gra}}(\hat{f}|V'_k). \tag{A7}$$

Here, the conversion for the three terms $\mathcal{E}^i_{\text{flo}}$, $\mathcal{E}^i_{\text{seg}}$ and $\mathcal{E}^{pc}_{\text{reg}}$ is exact, *i.e.*, those terms in $\mathcal{E}_{\text{top}}$ and corresponding terms in $\mathcal{A}$ yield the same energies as long as $f \equiv \hat{f}$. Next, we explain why these three conversions are exact and why the conversion for the two remaining terms are approximate.

Exact Conversion of Flow and Cosegmentation Data Terms

Exactness for the unary terms $\mathcal{E}^i_{\text{flo}}$ and $\mathcal{E}^i_{\text{seg}}$ in Eqs. (3) and (5) is shown in the same way. Notice that nodes in V_{k+1} are always obtained by merging nodes of V'_k, by following the rule of Eq. (19). Therefore, we can assume the domain Ω_i of each node $i \in V_{k+1}$ is the union of the domains of a connected component C_i of nodes in V'_k.

$$\Omega_i = \bigcup_{i' \in C_i} \Omega_{i'} \tag{A8}$$

Furthermore, from $f \equiv \hat{f}$ it holds that $f_i = \hat{f}_{i'}$ for $i \in V_{k+1}$ and $i' \in C_i$. Using these properties, a unary term $\mathcal{E}^i$ in $\mathcal{E}_{\text{top}}$ can be exactly converted to the form in $\mathcal{A}$ as follows. (Changes from previous equations are colored by blue).

$$\sum_{i \in V_{k+1}} \mathcal{E}^i(f_i) = \sum_{i \in V_{k+1}} \sum_{\mathbf{p} \in \Omega_i} \phi_{\mathbf{p}}(f_i) \tag{A9}$$

$$= \sum_{i \in V_{k+1}} \sum_{i' \in C_i} \sum_{\mathbf{p} \in \Omega_{i'}} \phi_{\mathbf{p}}(f_i) \tag{A10}$$

$$= \sum_{i \in V_{k+1}} \sum_{i' \in C_i} \sum_{\mathbf{p} \in \Omega_{i'}} \phi_{\mathbf{p}}(\hat{f}_{i'}) \tag{A11}$$

$$= \sum_{i' \in V'_k} \sum_{\mathbf{p} \in \Omega_{i'}} \phi_{\mathbf{p}}(\hat{f}_{i'}) \tag{A12}$$

$$= \sum_{i \in V'_k} \mathcal{E}^i(\hat{f}_i) \tag{A13}$$

Exact Conversion of Multi-layer Regularization Term

We perform a similar derivation for the multi-layer regularization term $\mathcal{E}_{\text{reg}}^{pc}$ in Eq. (10). From Figures 3 (c) and (d), we can see that each of the parent-child edges $(p, c) \in E_{k+1}^{\text{pc}}$ in the top layer of G^{k+1} has a corresponding edge $(p', c) \in E_{k+1}^{\text{pc}'}$ in $\hat{G}^{k+1}$ that has the same child c. Furthermore, for each one of those edges, $\mathbf{T}_p(\mathbf{p}) = \mathbf{T}_{p'}(\mathbf{p})$ and $\alpha_p = \alpha_{p'}$, since $f \equiv \hat{f}$. Therefore, we can exactly convert $\mathcal{E}_{\text{reg}}^{pc}$ in $\mathcal{E}_{\text{top}}$ to the form in $\mathcal{A}$ as follows.

$$\sum_{(p,c)\in E_{k+1}^{\text{pc}}} w_{pc}\, \mathcal{E}_{\text{reg}}^{pc}(f_p, f_c) = \sum_{(p,c)\in E_{k+1}^{\text{pc}}} w_{pc} \left[\lambda_{\text{pc1}} \min\{\alpha_p, \alpha_c\}\psi^{pc}(\mathbf{c}_c) + \lambda_{\text{pc2}}|\alpha_p - \alpha_c|\right] \tag{A14}$$

$$= \sum_{(p,c)\in E_{k+1}^{\text{pc}}} |\Omega_c| \left[\lambda_{\text{pc1}} \min\{\alpha_p, \alpha_c\} \min\{\|\mathbf{T}_p(\mathbf{c}_c) - \mathbf{T}_c(\mathbf{c}_c)\|_2, \tau_{\text{pc}}\} + \lambda_{\text{pc2}}|\alpha_p - \alpha_c|\right] \tag{A15}$$

$$= \sum_{(p',c)\in E_{k+1}^{\text{pc}'}} |\Omega_c| \left[\lambda_{\text{pc1}} \min\{\alpha_{p'}, \alpha_c\} \min\{\|\mathbf{T}_{p'}(\mathbf{c}_c) - \mathbf{T}_c(\mathbf{c}_c)\|_2, \tau_{\text{pc}}\} + \lambda_{\text{pc2}}|\alpha_{p'} - \alpha_c|\right] \tag{A16}$$

$$= \sum_{(p,c)\in E_{k+1}^{\text{pc}'}} w_{pc}\, \mathcal{E}_{\text{reg}}^{pc}(\hat{f}_p, \hat{f}_c) \tag{A17}$$

Approximate Conversion of Spatial Regularization Term

For the spatial regularization term $\mathcal{E}_{\text{reg}}^{st}$ in Eq. (6), we split it into two parts.

$$\sum_{(s,t)\in E'_k} w_{st}\, \mathcal{E}_{\text{reg}}^{st}(\hat{f}_s, \hat{f}_t) = \lambda_{\text{st1}} \sum_{(s,t)\in E'_k} w_{st}\, \min\{\alpha_s, \alpha_t\} \sum_{\mathbf{p}\in B_{st}} \psi^{st}(\mathbf{p})/|B_{st}| + \lambda_{\text{st2}} \sum_{(s,t)\in E'_k} w_{st}\, |\alpha_s - \alpha_t|. \tag{A18}$$

Here, the first and second parts evaluate flow and segmentation smoothness, respectively. We can show exact conversion for the segmentation smoothness part. To show this, we classify the edges of E'_k in $\hat{G}^{k+1}$ into two types: Type A) edges $(s', t') \in A$ across two different components $s' \in C_s$ and $t' \in C_t$. Type B) edges $(s'', t'') \in B$ within the same component $s'', t'' \in C_i$. Notice that $\mathcal{E}_{\text{reg}}^{st}(f_s, f_t) = 0$ for Type A edges, because $f_s = f_t$ holds in the same component. We now derive exact conversion for the segmentation smoothness part as follows.

$$\sum_{(s,t)\in E_{k+1}} w_{st}\, |\alpha_s - \alpha_t| = \sum_{(s,t)\in E_{k+1}} \left[\sum_{(s',t')\in A_{st}} w_{s't'}\right] |\alpha_s - \alpha_t| \tag{A19}$$

$$= \sum_{(s,t)\in E_{k+1}} \sum_{(s',t')\in A_{st}} w_{s't'}\, |\alpha_{s'} - \alpha_{t'}| \tag{A20}$$

$$= \sum_{(s',t')\in A} w_{s't'}\, |\alpha_{s'} - \alpha_{t'}| \tag{A21}$$

$$= \sum_{(s',t')\in A} w_{s't'}\, |\alpha_{s'} - \alpha_{t'}| + \sum_{(s'',t'')\in B} w_{s''t''}\, |\alpha_{s''} - \alpha_{t''}| \tag{A22}$$

$$= \sum_{(s,t)\in E'_k} w_{st}\, |\alpha_s - \alpha_t| \tag{A23}$$

Here, $w_{st} = \sum w_{s't'}$ in Eq. (A19) is from Eq. (14), but the definition of (s', t') can be equivalently replaced as Type A edges $(s', t') \in A_{st}$ where $s' \in C_s$ and $t' \in C_t$. Equation (A20) is from $f \equiv \hat{f}$, where it holds that $\alpha_i = \alpha'_i$ for $i \in V_{k+1}$ and $i' \in C_i$.

In contrast, the conversion of the flow smoothness part in Eq. (A18) is not always exact. However, the pixel locations $\mathbf{p}$ where the flow difference penalties $\psi^{st}(\mathbf{p})$ actually occur are the same in $\mathcal{E}_{\text{top}}$ and $\mathcal{A}$. Furthermore, the total costs of the flow

smoothness part are equally bounded by $\sum_{(s,t)\in E_{k+1}} \lambda_{\text{st}1} w_{st} \tau_{\text{st}}$ in both $\mathcal{E}_{\text{top}}$ and $\mathcal{A}$. Thus, Eq. (A18) is a good approximation for the spatial regularization term.

Approximate Conversion of Graph Validity Term

To derive an approximation $\mathcal{E}^{k+1}_{\text{gra}}(\hat{f}|V'_k)$ for the graph validity term $\mathcal{E}^{k+1}_{\text{gra}}(V_{k+1})$ in Eq. (11), we need to deal with two issues. 1) We need to convert variables from the node structure V_{k+1} in $\mathcal{E}_{\text{top}}$ to the labeling $\hat{f}$ on V'_k in $\mathcal{A}$. 2) The approximation function must be pairwise submodular energies for allowing graph cut based optimization.

For the first issue, we apply the variable conversion of Eq. (19) and regard V_{k+1} as a function $V_{k+1}(\hat{f})$ that represents a set of connected components C_i of nodes in V'_k assigned the same label. Thus, $\mathcal{E}^{k+1}_{\text{gra}}(V_{k+1})$ is converted to a function of $\hat{f}$ as follows.

$$\mathcal{E}^{k+1}_{\text{gra}}(V_{k+1}) = \lambda_{\text{nod}} \beta^{k+1} |V_{k+1}| - \lambda_{\text{col}} \sum_{i\in V_{k+1}} \sum_{\mathbf{p}\in\Omega_i} \ln P(\mathbf{I}_\mathbf{p}|\boldsymbol{\theta}^i) \tag{A24}$$

$$= \lambda_{\text{nod}} \beta^{k+1} |V_{k+1}(\hat{f})| - \lambda_{\text{col}} \sum_{i\in V_{k+1}} \sum_{\mathbf{p}\in\Omega_i} \ln P(\mathbf{I}_\mathbf{p}|\boldsymbol{\theta}^{C_i}) \tag{A25}$$

$$= \lambda_{\text{nod}} \beta^{k+1} |V_{k+1}(\hat{f})| - \lambda_{\text{col}} \sum_{i'\in V'_k} \sum_{\mathbf{p}\in\Omega_{i'}} \ln P(\mathbf{I}_\mathbf{p}|\boldsymbol{\theta}^{C_i}) \tag{A26}$$

Here, $|V_{k+1}(\hat{f})|$ is the count of the components defined by the labeling $\hat{f}$, and $\boldsymbol{\theta}^{C_i}$ is the color distribution within the region of a component C_i that $i' \in V'_k$ belongs to. The fact that the computation of both $|V_{k+1}(\hat{f})|$ and $\boldsymbol{\theta}^{C_i}$ involves regional (higher-order) information of $\hat{f}$ raises the second issue.

To deal with the second issue of higher-order terms, we relax the connectivity of $|V_{k+1}(\hat{f})|$ and treat it as the count of unique labels $\hat{f}_i$ in $i \in V'_k$ without considering their spatial connections.

$$|V_{k+1}(\hat{f})| \simeq \sum_{L\in\{\text{all labels}\}} \delta_L(\hat{f}), \tag{A27}$$

where $\delta_L(\hat{f}) = 1$ if $\exists i \in V'_k : \hat{f}_i = L$; otherwise $\delta_L(\hat{f}) = 0$. In this manner, $|V_{k+1}(\hat{f})|$ becomes *label costs* [2] of $\hat{f}$, and the formulation of Eq. (A26) is the same as that of multi-region segmentation of [2]. In their model fitting approach, the label costs are optimized as pairwise submodular terms under alpha expansion moves with additional auxiliary variables. Our optimization approach using local expansion moves allows the same strategy. Furthermore, the distribution $\boldsymbol{\theta}^{C_i}$ is treated as a label $\boldsymbol{\theta}_i$ given by $\hat{f}_i$, rather than a value computed from C_i. Thus, the likelihood terms in Eq. (A26) are approximated as unary potentials as follows.

$$-\sum_{i'\in V'_k} \sum_{\mathbf{p}\in\Omega_{i'}} \ln P(\mathbf{I}_\mathbf{p}|\boldsymbol{\theta}^{C_i}) \simeq \sum_{i\in V'_k} \mathcal{E}^i_{\text{gra}}(\hat{f}_i) \tag{A28}$$

where $\mathcal{E}^i_{\text{gra}}(\hat{f}_i)$ evaluates the given distribution label $\boldsymbol{\theta}_i$ included in $\hat{f}_i$ as

$$\mathcal{E}^i_{\text{gra}}(\hat{f}_i) = -\sum_{\mathbf{p}\in\Omega_i} \ln P(\mathbf{I}_\mathbf{p}|\boldsymbol{\theta}_i). \tag{A29}$$

Note that the energy conversion is unnecessary for the graph terms in $\mathcal{E}(\hat{f}|G^k)$, because those terms are constant with the fixed G^k. Likewise, it is unnecessary in the whole process of the top-down labeling refinement phase.

Consequently, $\hat{f}$ becomes the following labeling on $\hat{G}^{k+1}$.

$$\hat{f}_i = \begin{cases} (\mathbf{T}_i, \alpha_i, \boldsymbol{\theta}_i) & \text{if } i \in V'_k \\ (\mathbf{T}_i, \alpha_i) & \text{if } i \in V_l \ (0 \le l \le k) \end{cases}. \tag{A30}$$

The distribution label $\boldsymbol{\theta}_i$ of $i \in V'_k$ is initialized as the color distribution of the region Ω_i. Except for the cross-view proposer, the proposal generation for distribution labels is essentially the same as that of other labels $(\mathbf{T}, \alpha)$. The expansion and perturbation proposers simply copy the current label $\boldsymbol{\theta}_i$ of the target node i as a candidate. The average proposer generates candidates as the weighted sum of two distributions $w_i\boldsymbol{\theta}_i + w_j\boldsymbol{\theta}_j$. The cross-view proposer generates a candidate as the distribution within the region Ω_i of the target node i.

C. Initiailzation of Color Models

Here, we explain the implementation details of the initialization of color models $\{\boldsymbol{\theta}^F, \boldsymbol{\theta}^B\}$ omitted in Section 5.3.

Geodesic Distance

We first compute a geodesic distance map for each of the input images. At every pixel $\mathbf{p}$ we compute the shortest geodesic distance to any of the image boundary pixels $\mathbf{q} \in B$:

$$D(\mathbf{p}) = \min_{\mathbf{q} \in B} d(\mathbf{p}, \mathbf{q}), \tag{A31}$$

where $d(\mathbf{p}, \mathbf{q})$ is the geodesic distance between two pixels $\mathbf{p}$ and $\mathbf{q}$ define as

$$d(\mathbf{p}, \mathbf{q}) = \min_{s \in \mathcal{P}} \sum_{k=1}^{|s|-1} \|\mathbf{I}(\mathbf{s}(k+1)) - \mathbf{I}(\mathbf{s}(k))\|_2. \tag{A32}$$

Here, $\mathcal{P}$ is the set of all paths joining $\mathbf{p}$ and $\mathbf{q}$. The approximate computation of $D(\mathbf{p})$ is efficiently implemented using a linear-order algorithm of [11].

We further normalize the value range of the geodesic distance map by

$$\bar{D}(\mathbf{p}) = e^{-D(\mathbf{p})^2/\gamma}. \tag{A33}$$

The parameter γ is given as $\gamma = \eta\sigma^2$ where $\sigma = E[\|\mathbf{I}(\mathbf{p}) - \mathbf{I}(\mathbf{q})\|_2]$ is computed as the expectation over all spatial neighbors $(\mathbf{p}, \mathbf{q})$, and η is set to 20 in our implementation. The values of $1 - \bar{D}(\mathbf{p})$ are visualized in the right part of Figure 4.

Seeds and Initial Mask Creation for GrabCut

Secondly, we compute seeds and initial masks of foreground and background as input to GrabCut [8]. The seeds of foreground and background regions give constant unary likelihoods. The initial masks are used to initialize the color distributions used in GrabCut. We compute these regions using the ratio values and the geodesic distance as follows.

As explained in Section 5.3, we have three ratio values $\{r_1, r_2, r_3\}$ at each pixel computed from the three levels of the image pyramid. For each level, we normalize the ratio values to be in the range of $[0, 1]$ using the minimum and maximum ratio values. After the layerwise normalization, we integrate the three ratio values to obtain a single value as $r = r_1r_2r_3 + (1-r_1)r_2r_3 + r_1(1-r_2)r_3 + r_1r_2(1-r_3)$. We then create the foreground / background seeds and foreground / background masks as regions where $r < 0.05$, $r > 0.95$, $r < 0.70$ and $r > 0.85$, respectively. The regions of foreground seed and mask are further reduced if the geodesic distance is $\bar{D}(\mathbf{p}) > 0.5$.

In our implementation, the color likelihood terms of GrabCut are implemented by 64^3 bins of RGB color histograms with a weight coefficient of 1. The pixels in the foreground/background seeds are assigned a constant likelihood value of 10. Using the geodesic distance in Eq. (A33), we also add background likelihood values of $10\bar{D}(\mathbf{p})$. For efficiency, we use the superpixel nodes of V_1 during this step and reuse them again in our main algorithm. Finally, we obtain estimated color models $\{\boldsymbol{\theta}^F, \boldsymbol{\theta}^B\}$ of an image after a few iterations of GrabCut. We perform this computation for each of the two images.

D. Submodularity

As discussed in [10, 9] the submodularity condition of local expansion move energies in Eq. (20) is the same as that of conventional alpha expansion moves [1]. To prove that our energy is submodular under expansion moves, we need to show that our pairwise regularization terms $\mathcal{E}_{\text{reg}}^{sp}$ and $\mathcal{E}_{\text{reg}}^{pc}$ in Eqs. (6) and (10) are submodular. To simplify discussions, we rewrite these terms as a pairwise function, as follows.

$$\phi(\mathbf{x},\mathbf{y}) = \min\{x,y\}\psi(\mathbf{x},\mathbf{y}) + \lambda|x-y|. \tag{A34}$$

Here, $\lambda \geq 0$ is a scalar weight, a bold $\mathbf{x}$ denotes a label vector of $(\mathbf{T},\alpha)$ while a non-bold x denotes its scalar alpha label $\alpha \in [0,1]$. The two terms $\mathcal{E}_{\text{reg}}^{sp}$ and $\mathcal{E}_{\text{reg}}^{pc}$ can be expressed in this form by properly defining $\psi(\mathbf{x},\mathbf{y})$. Using this notation we prove the following two lemmas.

Lemma 1 *If $\psi(,)$ satisfies the following three conditions for any $\mathbf{x},\mathbf{y},\mathbf{z}$*

$$0 \leq \psi(\mathbf{x},\mathbf{y}) \leq \tau, \tag{A35}$$

$$\psi(\mathbf{x},\mathbf{x}) = 0, \tag{A36}$$

$$\psi(\mathbf{x},\mathbf{y}) + \psi(\mathbf{z},\mathbf{z}) \leq \psi(\mathbf{x},\mathbf{z}) + \psi(\mathbf{z},\mathbf{y}), \tag{A37}$$

and if

$$\tau \leq 2\lambda, \tag{A38}$$

then $\phi(\mathbf{x},\mathbf{y})$ is submodular under expansion moves, i.e.*, it satisfies the following submodularity condition of expansion moves [1, 6]:*

$$\phi(\mathbf{x},\mathbf{y}) + \phi(\mathbf{z},\mathbf{z}) \leq \phi(\mathbf{x},\mathbf{z}) + \phi(\mathbf{z},\mathbf{y}). \tag{A39}$$

Proof.

Notice that $\phi(\mathbf{z},\mathbf{z}) = 0$. Using this and assuming $x \geq y$ without loss of generality, Eq. (A39) can be expressed as

$$\min\{x,z\}\psi(\mathbf{x},\mathbf{z}) + \lambda|x-z| + \min\{z,y\}\psi(\mathbf{z},\mathbf{y}) + \lambda|z-y| - y\psi(\mathbf{x},\mathbf{y}) - \lambda(x-y) \geq 0. \tag{A40}$$

The proof for the above inequity is divided into the following three cases depending on z.

Case 1 where $x \geq y \geq z \geq 0$. We show in this case that

$$\text{Eq. (A40, left)} = z\psi(\mathbf{x},\mathbf{z}) + \lambda(x-z) + z\psi(\mathbf{z},\mathbf{y}) + \lambda(y-z) - y\psi(\mathbf{x},\mathbf{y}) - \lambda(x-y) \tag{A41}$$

$$= z\Big[\psi(\mathbf{x},\mathbf{z}) + \psi(\mathbf{z},\mathbf{y})\Big] - y\psi(\mathbf{x},\mathbf{y}) + 2\lambda(y-z) \tag{A42}$$

$$\geq z\psi(\mathbf{x},\mathbf{y}) - y\psi(\mathbf{x},\mathbf{y}) + 2\lambda(y-z) \tag{A43}$$

$$= (y-z)\Big[2\lambda - \psi(\mathbf{x},\mathbf{y})\Big] \tag{A44}$$

$$\geq (y-z)\Big[2\lambda - \tau\Big] \tag{A45}$$

$$\geq 0. \tag{A46}$$

Case 2 where $x \geq z \geq y \geq 0$. Similarly, we show that

$$\begin{aligned}
\text{Eq. (A40, left)} &= z\psi(\mathbf{x},\mathbf{z}) + \lambda(x-z) + y\psi(\mathbf{z},\mathbf{y}) + \lambda(z-y) - y\psi(\mathbf{x},\mathbf{y}) - \lambda(x-y) && \text{(A47)}\\
&= z\psi(\mathbf{x},\mathbf{z}) + y\psi(\mathbf{z},\mathbf{y}) - y\psi(\mathbf{x},\mathbf{y}) && \text{(A48)}\\
&\geq y\Big[\psi(\mathbf{x},\mathbf{z}) + \psi(\mathbf{z},\mathbf{y}) - \psi(\mathbf{x},\mathbf{y})\Big] && \text{(A49)}\\
&\geq 0. && \text{(A50)}
\end{aligned}$$

Case 3 where $z \geq x \geq y \geq 0$. Finally, we show that

$$\begin{aligned}
\text{Eq. (A40, left)} &= x\psi(\mathbf{x},\mathbf{z}) + \lambda(z-x) + y\psi(\mathbf{z},\mathbf{y}) + \lambda(z-y) - y\psi(\mathbf{x},\mathbf{y}) - \lambda(x-y) && \text{(A51)}\\
&= x\psi(\mathbf{x},\mathbf{z}) + y\psi(\mathbf{z},\mathbf{y}) - y\psi(\mathbf{x},\mathbf{y}) + 2\lambda(z-x) && \text{(A52)}\\
&\geq y\Big[\psi(\mathbf{x},\mathbf{z}) + \psi(\mathbf{z},\mathbf{y}) - \psi(\mathbf{x},\mathbf{y})\Big] + 2\lambda(z-x) && \text{(A53)}\\
&\geq 0. && \text{(A54)}
\end{aligned}$$

Lemma 2 *If $\psi(\mathbf{x},\mathbf{y})$ is given by a form of the truncated Euclidean distance as*

$$\psi(\mathbf{x},\mathbf{y}) = \min\{\|\mathbf{x}-\mathbf{y}\|_2, \tau\}, \tag{A55}$$

then $\psi(\mathbf{x},\mathbf{y})$ satisfies the aforementioned three conditions of Eqs. (A35) – (A37).

Proof.

The first and second conditions are obvious. We can also show the third condition as follows.

$$\begin{aligned}
\psi(\mathbf{x},\mathbf{y}) + \psi(\mathbf{z},\mathbf{z}) &= \psi(\mathbf{x},\mathbf{y}) && \text{(A56)}\\
&= \min\{\|\mathbf{x}-\mathbf{y}\|_2, \tau\} && \text{(A57)}\\
&= \min\{\|(\mathbf{x}-\mathbf{z}) - (\mathbf{y}-\mathbf{z})\|_2, \tau\} && \text{(A58)}\\
&\leq \min\{\|\mathbf{x}-\mathbf{z}\|_2 + \|\mathbf{y}-\mathbf{z}\|_2, \tau\} && \text{(A59)}\\
&\leq \min\{\|\mathbf{x}-\mathbf{z}\|_2, \tau\} + \min\{\|\mathbf{y}-\mathbf{z}\|_2, \tau\} && \text{(A60)}\\
&= \psi(\mathbf{x},\mathbf{z}) + \psi(\mathbf{z},\mathbf{y}) && \text{(A61)}
\end{aligned}$$

The above two lemmas directly derive the submodularity for the parent-children term $\mathcal{E}^{pc}_{\text{reg}}$ using substitutions $\lambda = \lambda_{\text{pc2}}$ and $\tau = \lambda_{\text{pc1}}\tau_{\text{pc}}$. By slightly modifying Eq. (A55) for the spatial term $\mathcal{E}^{st}_{\text{reg}}$, it can also be shown to be submodular where $\lambda = \lambda_{\text{st2}}$ and $\tau = \lambda_{\text{st1}}\tau_{\text{st}}$.

E. Tuning Hyper Parameters

We explain our strategy of tuning parameters. Since the graph term is independent of the labeling, we start with a simple energy function consisting of only the graph term. We set $\lambda_{\text{col}} = 1$ and tune λ_{nod} so that $|V_1| \simeq 2|V_2|$ in the obtained graph. We then use the single layer model and tune parameters of the flow (λ_{flo}, τ_D) and segmentation (λ_{seg}) data terms and spatial smoothness term (λ_{st1}, λ_{st2}, τ_{st}). While checking segmentation quality, we tune λ_{seg} at around λ_{col} and λ_{st2} at around 50 (the default setting in GrabCut [8]). The remaining flow-related parameters are tuned by checking flow quality. We finally use the hierarchical model and tune the parameters (λ_{pc1}, λ_{pc2}, τ_{pc}) of the multi-layer regularization.

Table A1. Benchmark results (**without flipped images**). FAcc is flow accuracy rate for an error threshold of 5 pixels in a normalized scale. SAcc is segmentation accuracy by intersection-over-union ratios. SAcc scores ($\star$) of optic flow mothods are computed by post-processing using left right consistency check.

Optic flow / cosegment. Methods	FG3DCar		JODS		PASCAL	
	FAcc	SAcc	FAcc	SAcc	FAcc	SAcc
Ours	**0.837**	**0.756**	**0.665**	0.521	**0.754**	**0.659**
Our single layer ([10])	0.734	**0.757**	0.525	0.515	0.649	0.626
SIFT Flow [7]	0.640	(0.420)	0.582	(0.257)	0.695	(0.468)
DSP [5]	0.492	(0.285)	0.517	(0.227)	0.590	(0.364)
DFF [12]	0.498	(0.328)	0.330	(0.213)	0.333	(0.251)
Faktor and Irani [3]	–	0.688	–	**0.549**	–	0.486
Joulin *et al.* [4]	–	0.461	–	0.332	–	0.411

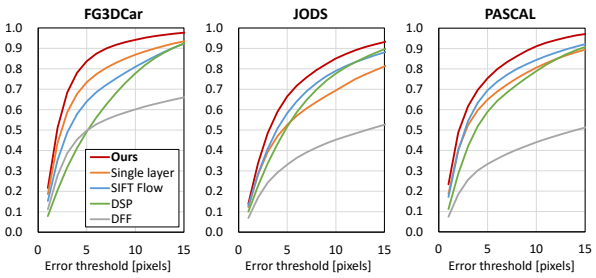

Figure A5. Average flow accuracies evaluated by endpoint errors with varying thresholds (**without flipped images**). Ours always shows best scores. Similarly to Figure 5, our method shows always best scores.

F. Dataset

In this section, we show more examples and report some statistics of our dataset. Our dataset comprises of 400 image pairs divided into three groups – **FG3DCar** contains 195 image pairs of vehicles. **JODS** contains 81 image pairs of airplanes, horses, and cars. **PASCAL** contains 124 image pairs of bicycles, motorbikes, buses, cars, trains. The charts in Figure A1 show the number of image pairs in each subcategory of JODS and PASCAL. Figures A2–A4 show examples of image pairs from FG3DCar, JODS and PASCAL, respectively. Notice that JODS and PASCAL contain some horizontally flipped image pairs, *i.e.*, one image requires a mirror reflection prior to alignment. The numbers of such flipped image pairs included in each group are follows. FG3DCar: 2 pairs (1 %). JODS: 9 pairs (11 %). PASCAL: 48 pairs (39 %).

G. Benchmark Scores without Flipped Images

As mentioned in the previous section, our dataset contains flipped image pairs. Since our method and others do not explicitly handle such image pairs, they fail to find correspondence for them. Therefore, we also evaluate accuracy scores similar to Table 1 and Figure 5 but excluding flipped image pairs from the evaluation. We show the average scores for the three groups in Table A1, and the plots of average flow accuracies with varying thresholds in Figure A5. We observe similar trends between scores with and without flipped image pairs.

References

[1] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 23(11):1222–1239, 2001.

[2] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. *Int'l Journal of Computer Vision*, 96(1):1–27, 2012.

[3] A. Faktor and M. Irani. Co-segmentation by composition. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pages 1297–1304, 2013.

[4] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[5] J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2307–2314, 2013.

[6] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 26(2):147–159, 2004.

[7] C. Liu, J. Yuen, and A. Torralba. SIFT Flow: Dense Correspondence across Scenes and Its Applications. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 33(5):978–994, 2011.

[8] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. on Graph.*, 23(3):309–314, 2004.

Figure A1. Subcategories of JODS and PASCAL.

Figure A2. Examples of FG3DCar

Figure A3. Examples of JODS

Figure A4. Examples of PASCAL

[9] T. Taniai, Y. Matsushita, and T. Naemura. Graph cut based continuous stereo matching using locally shared labels. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1613–1620, 2014.

[10] T. Taniai, Y. Matsushita, Y. Sato, and T. Naemura. Continuous Stereo Matching Using Local Expansion Moves. arXiv:1603.08328, http://arxiv.org/abs/1603.08328, 2016.

[11] P. J. Toivanen. New Geodesic Distance Transforms for Gray-scale Images. *Pattern Recogn. Lett.*, 17(5):437–450, 1996.

[12] H. Yang, W. Lin, and J. Lu. DAISY filter flow: A generalized discrete approach to dense correspondences. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3406–3413, 2014.