Optical Flow Estimation on Coarse-to-Fine Region-Trees using Discrete Optimization

Cheng Lei and Yee-Hong Yang
Department of Computing Science
University of Alberta, Edmonton, AB, Canada
{clei,yang}@cs.ualberta.ca

Abstract

In this paper, we propose a new region-based method for accurate motion estimation using discrete optimization. In particular, the input image is represented as a tree of over-segmented regions and the optical flow is estimated by optimizing an energy function defined on such a region-tree using dynamic programming. To accommodate the sampling-inefficiency problem intrinsic to discrete optimization compared to the continuous optimization based methods, both spatial and solution domain coarse-to-fine (C2F) strategies are used. That is, multiple region-trees are built using different over-segmentation granularities. Starting from a global displacement label discretization, optical flow estimation on the coarser level region-tree is used for defining region-wise finer displacement samplings for finer level region-trees. Furthermore, cross-checking based occlusion detection and correction and continuous optimization are also used to improve accuracy. Extensive experiments using the Middlebury benchmark datasets have shown that our proposed method can produce top-ranking results.

1. Introduction

As an active research topic for many years, the goal of optical flow estimation is to recover a dense 2D vector field encoding scene object motion or camera motion as displacements between corresponding pixels in consecutive images. For a more extensive review, the reader can refer to some previous surveys such as [19, 20].

Following the seminal work of Horn and Schunck [2], most state-of-the-art algorithms formulate the optical flow estimation as an energy minimization problem [3-15], for which the variational-calculus based computation framework has been the top-performing method and gained much popularity in the research community [3-6]. However, such optimization schemes based on continuous mathematics often suffer the problems of over-smoothing due to their restricted convex flow smoothness regularizations. Also due to their gradient-descent based minimum searching, the optimization could be trapped by local minima, which result in poor performance for sharp motion discontinuities and for large motion displacements [10].

On the other hand, a similar energy-minimization computational framework as above also dominates in the top-ranking disparity/depth estimation algorithms. However, in contrast, discrete optimization schemes such as graph-cuts, belief propagation and dynamic programming have gained more popularity over the continuous counterparts due to their better ability in optimizing non-convex energy functions.

Then naturally it leads one to wonder if such discrete optimization schemes are also applicable to optical flow estimation which bears commonalities with the stereo matching problem. To this end, considerable efforts [7-11] have been devoted recently in enabling discrete optimization schemes for optical flow estimation and promising results have been reported using the Middlebury quantitative evaluation and benchmarking datasets [1]. Moreover, compared to pixel-based stereo matching approaches, region based ones have demonstrated their superior capability in handling texture-less regions and occlusions, which are also problematic issues in optical flow estimation. Surprisingly, there are only very few works that have incorporated color segmentation information for better optical flow estimation. So in this paper, we propose a new coarse-to-fine region tree based optical flow estimation method which combines the proven advantages of discrete optimization and region based image representation. In the following, we first review some previous related work in Section 2. Then in Sections 3 and 4, we give an overview first and then, elaborate our proposed approach. In Section 5, we show that the proposed approach can achieve superior performance based on the Middlebury optical flow evaluation. Finally, we conclude our paper in Section 6.

2. Related work

Several previous attempts have applied discrete optimization schemes in optical flow estimation. In general, the original optical flow problem is mapped into a labeling problem through discretization and then a well-known discrete optimization scheme such as graph-cuts [7] or belief-propagation [8] is adapted to

2009 IEEE 12th International Conference on Computer Vision (ICCV)
978-1-4244-4419-9/09/$25.00 ©2009 IEEE

finding the best label assignment for all the labeling targets (matching primitives) such as pixels, regions or layers, from which the final optical flow field is induced by mapping the labels back to displacement vectors.

Based on whether or not discretization is directly done in the flow solution space, we can roughly classify such methods as direct [7-9, 12] and indirect [10-11, 15] discretization based methods. In direct discretization based methods, the labels are a direct discrete sampling of the final 2D displacement search space. That is, each label corresponds to a sampled 2D displacement vector. While in indirect discretization based methods, no displacement discrete sampling is done.

Many recently proposed methods with very promising performance also adapt the discrete optimization schemes [10-12]. Specifically, in the fusion-flow method [10], the pixel-wise label sets are locally created from a set of candidate solutions obtained by running different continuous flow algorithms or the same algorithm using different parameter settings. Then graph-cuts optimization is used to find the best label assignment for fusing candidate solutions. A similar fusion idea has also been investigated in [11]. The original minimization problem is formulated as a series of binary sub-problems, each of which can be solved iteratively via the extended discrete graph-cuts with alpha-expansion method that facilitates large energy minimization moves. Similar to [10], the set of candidate displacement vectors to be fused have to be provided by standard continuous optical flow algorithms. Thus the success of both methods is largely dependent on the quality of the initial solution. Another piece of related work is presented in [12], in which a framework based on a dynamic, discrete MRF is proposed for morphing images using a grid of control points. Discrete MRF optimization is used to iteratively and accumulatively optimize the displacement vectors at the control points from which the dense optical flow field is derived based on the influence functions.

The promising performance as demonstrated using the Middlebury benchmark database of all of the above mentioned recent attempts suggests that discrete optimization has great potential in optical flow estimation. However, in addition to the optimization framework, the image representation can also play an important role in the performance. In particular, region based representation has shown unique advantages over pixel based representation in stereo matching [17, 18]. So it is intuitive to expect similar applicability and advantages of region based representation in optical flow estimation.

Some efforts [13-16] have also been made in this regard. In particular, in [14], a method is proposed that can jointly segment consecutive frames into small regions of consistent size and compute the optical flow based on statistical modeling of an image pair using constraints based on appearance and motion. Bidirectional motion is estimated using spatial coherence and color similarity between segmented regions under the translational motion model. In [15], image segmentation and graph-cuts optimization are incorporated to tackle the optical flow problem using a layered model. Each region is first assigned with an affine motion model from sparse correspondences. Motion layers are extracted by grouping regions with similar rigid motions and by identifying the dominant ones. Then as an indirect discretization based method, an energy function measuring the quality of label assignments of regions and pixels to layers is minimized via graph-cuts. Although very promising results have been obtained, the assumption on the existence of dominant rigid motion layers limits its applications. Different from [14, 15], [16] uses the segmented color regions as soft constraints in the affine motion model in the classic variational optical flow framework as regularization instead of as matching primitives. To avoid over-regularization on non-rigid motions, a confidence map encoding the fitness of the affine region motion model is used.

Despite their differences, all of the above efforts of incorporating segmentation information for optical flow estimation have commonly observed significant performance improvements in handling texture-less regions and in preserving sharp motion discontinuities.

Our work is closely related to [18], in which stereo matching is done by optimizing an energy function defined on a minimum spinning tree of over-segmented image regions using dynamic programming. The advantage of using such a region-tree based representation over using the region graph as in [14, 17] is that under the assumption that depth discontinuities coincide with intensity discontinuities, the number of edges that cross depth discontinuities, i.e., violate the smoothness constraint, can be minimized. This can result in "smarter" smoothness enforcement [22,23] in subsequent optimization. Furthermore, the cycle-less region tree structure also enables us to use simpler or more efficient optimization methods such as dynamic programming. In this paper, by taking advantage of commonalities between optical flow estimation and stereo matching, we extend [18] and propose a new coarse-to-fine region-tree framework for optical flow estimation.

3. Propose algorithm

3.1. Problem formulation and notations

Our paper is direct discretization based and formulates the optical flow estimation as a discrete energy minimization problem. In particular, two consecutive input images $\boldsymbol{I}_t$ and $\boldsymbol{I}_{t+1}$ are represented by a spatial structure Ω of a set of matching primitives $\mathcal{P}$ spanning the whole image. Then the optical flow field $(\boldsymbol{U}, \boldsymbol{V})$ from image $\boldsymbol{I}_t$ to $\boldsymbol{I}_{t+1}$ is recovered through finding an optimal

labeling $\mathbb{L}(\Omega)$ which assigns each primitive $p \in \mathcal{P}_{I_t}$ a label $\ell \in \mathcal{L}$, where $(\boldsymbol{U}, \boldsymbol{V})$ denote the horizontal and vertical components of the displacement vector field, respectively. Such an $\mathbb{L}(\Omega)$ is found by minimizing an energy function $E(\mathbb{L}(\Omega))$ with the general form of

$$E(\boldsymbol{L}(\Omega)) = E_{data}(\mathbb{L}(\Omega)) + \lambda \cdot E_{smooth}(\mathbb{L}(\Omega)) \quad (1)$$

where E_{data} represents the data term and E_{smooth} the smoothness term.

Similar to many previous works, the data term $E_{data}(\boldsymbol{L})$ measures the brightness matching error between two images correlated by a warping induced from the optical flow $(\boldsymbol{U}, \boldsymbol{V})$ corresponding to $\mathbb{L}$. And the smoothness term $E_{smooth}(\mathbb{L})$ enforces the piecewise smoothness regularity of the optical flow by penalizing spatial variance in the flow field $(\boldsymbol{U}, \boldsymbol{V})$. The positive constant λ gives the relative weight of the smoothness penalty.

Each primitive $p \in \mathcal{P}_{I_t}$ maintains its own displacement look-up-table (LUT) $\mathcal{D}_p$ which defines a bijection mapping of each label $\ell \in \mathcal{L}$ to the corresponding 2D displacement vector (u_ℓ, v_ℓ). By looking-up $\mathcal{D}_p$, the energy terms of the candidate label can be evaluated during optimization and the optical flow field $(\boldsymbol{U}, \boldsymbol{V})$ is induced from the resultant labeling $\mathbb{L}$.

3.2. Overview

Our proposed method is a region-based one and uses a new coarse-to-fine (C2F) paradigm. That is, as illustrated in Figure 1, multiple-level coarse-to-fine over-segmentations $\{S_l | l = 0, \ldots, M\}$ are done to the input images. For each segmentation level l, the over-segmented regions $\mathcal{R}_l$ form the corresponding primitive set $\mathcal{P}_l$ on which a spanning region-tree $\mathcal{T}_l$ is built as Ω_l.

Then starting from the coarsest segmentation level to the finest one, the corresponding labeling problem is solved through minimizing the energy function (1) defined on the region-tree $\mathcal{T}_l$ using dynamic programming (DP). The results of the coarser level region-tree are used by the finer level to refine the search range of motion displacements. Finally the resulting optical flow of the finest level is further smoothed using local continuous optimization. Also cross-checking based inconsistency detection can be optionally done to correct errors due to occlusions by similarly recovered optical flow $(\boldsymbol{U}', \boldsymbol{V}')$ from $\boldsymbol{I}_{t+1}$ to $\boldsymbol{I}_t$.

For better clarity, our proposed algorithm is summarized as follows.

Step 1: Build image pyramids and use downsized images to probe the initial displacement search ranges (Section 4.3)
Step 2: At each image pyramid level, over-segment image $\boldsymbol{I}_t$ using M different granularity constraints and build the corresponding M region-trees (Section 4.1)
Step 3: Iterate from segmentation level $l = 0$ to M
Step 4: In each iteration:
  (a) Setup the label-to-displacement LUT $\mathcal{D}_{r_l}$ for each region r_l in the current region-tree $\mathcal{T}_l$ (Section 4.2 and 4.3)
  (b) Evaluate "label space images" for all the hypothesized labels $\ell \in \mathcal{L}$. GPU is used for better efficiency in fast image interpolation (Section 4.4)
  (c) Run DP to optimize the corresponding energy function (7) and induce the optical flow field $(\boldsymbol{U}, \boldsymbol{V})_l$ from the resultant optimal labelling $\mathbb{L}(\mathcal{T}_l)$ (Section 4.4)
Step 5: Goto Step 3 if $l < M$, otherwise obtain the optical flow $(\boldsymbol{U}, \boldsymbol{V})$ from $\boldsymbol{I}_t$ to $\boldsymbol{I}_{t+1}$ at the current pyramid level
Step 5: (Optional) Recover the optical flow $(\boldsymbol{U}', \boldsymbol{V}')$ from $\boldsymbol{I}_{t+1}$ to $\boldsymbol{I}_t$ and perform cross-checking based correction (Section 4.5-a)
Step 6: Perform continuous optimization for smoothing (Section 4.5-b)
Step 7: Goto Step 2 if there is a finer scale image pyramid level

---

List 1: Workflow of our proposed algorithm

Please note that throughout this paper, we use the same scheme as that specified in the Middlebury Optical Flow site to color the optical flow as shown in Figure 1.

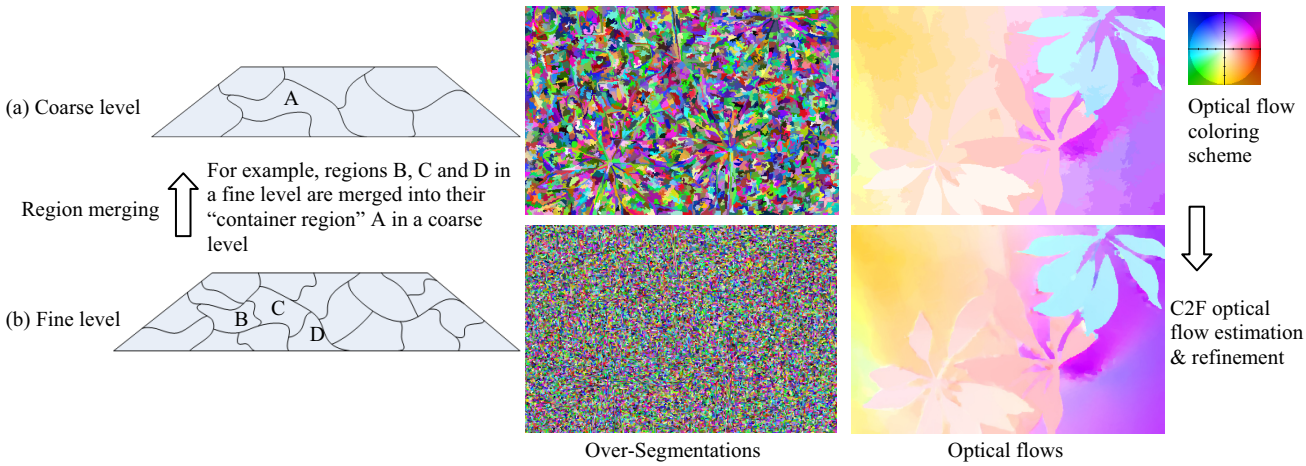


Figure 1: Two-Level coarse-2-fine over-segmentations of the dateset "Schefflera" and its optical flow recovery procedure.

4. Implementation details

4.1. Coarse-to-fine region trees

So far, two image representations are most often used in motion estimation -- the traditional pixel grid and the motion layer representations. The former one is simple but suffers from higher ambiguous matching possibility, while the main challenge of using the latter one is in difficulties of correct layer segmentation and layer motion parameterization without prior knowledge.

As a trade-off between enabling matching primitives to contain enough information with a large support area and reducing the risk of violating the parameterization assumption with a small support area, representations using over-segmented regions have shown great potentials in [14, 17, 18]. Such representations can reduce the computational complexity compared to pixel-based representations due to much fewer numbers of regions. Also given the smaller size of the regions, the chance that color segmentation errors propagate into the matching process is reduced compared to layer based representation.

These advantages motivate our new C2F region-tree based image representation. In particular, similar to [18], we apply mean-shift filtering [24] to the source image first and then a fusion process is iteratively performed to fuse most similar adjacent regions (pixels in the first iteration) into larger regions. By controlling the lower bound of the minimal region size, different granularity of image over-segmentations can be obtained. Such merging based process makes it possible to obtain efficiently multiple level over-segmentations in one single pass and guarantees that each larger region in the coarser level consists of exactly the same smaller regions that are in the larger region in its corresponding finer level segmentation. That is, each region $r \in \mathcal{R}_l$ in a coarser level l is composed of a group of adjacent regions $\in \mathcal{R}_k$ of finer segmentation level $k > l$, for each of which region r is its "container" region.

Then for each segmentation level, a region adjacency graph is first constructed with the edge weights indicating the dissimilarity between two adjacent regions. From it, a minimal spanning tree is extracted so that the sum of the remaining edge weights is minimized. Please refer to [18] for more details on building a region-tree for each segmentation level.

In this way, in the coarse level, larger regions make finding roughly correct matches easier and more robust so that the region-dependent search range of interest can be located quickly, while in the finer level, small-size regions are better at recovering subtle details via local range refinements. Just using large-size regions will make it difficult to capture small motion in an optical flow field, for which small-size regions or pixels are preferred. However, if the region size is too small, the disadvantages similar to using single pixels may prevail. Therefore this coarse-to-fine region-tree representation provides a tradeoff to get the best of pixel and layer based representations.

4.2. Displacement discretization

Using discrete optimization to recover essentially continuous optical flow requires discretization. As a direct discretization based method, the continuous 2D displacement solution space has to be quantized and mapped to a discrete set of labels. However, brute-force discretization usually suffers from the so-called "*discretization bottleneck*" problem, which means that the number of labels required for sampling the search ranges with fine enough precision could be too large for efficient optimization.

This problem is addressed using the above proposed coarse-to-fine region-tree representation. Specifically, at each level l, the displacement LUT $\mathcal{D}_{r_l}$ for each region $r_l \in \mathcal{R}_l$ is built by uniformly sampling the corresponding displacement search ranges $\left[u_{min}^{r_l}, u_{max}^{r_l}\right]$ and $\left[v_{min}^{r_l}, v_{max}^{r_l}\right]$ in both the horizontal and vertical directions with a sampling interval δ_l. At the coarsest level, the displacement search ranges of all regions are initialized globally with

$$u_{max}^{r_l} = v_{max}^{r_l} = -u_{min}^{r_l} = -v_{min}^{r_l} = \alpha \cdot \max(w, h) \quad (2)$$

wherein w and h are the width and height of the input images, respectively, and α is a positive constant. That is, we assume that the horizontal and vertical displacements each have upper bounds related to the image dimension. Then with the result of the last coarser l level known, the search range of a region $r_k \in \mathcal{R}_k$ at the finer level $k > l$ is setup based on its container region $r_l \in \mathcal{R}_l$ at level l. That is, suppose that the recovered displacement vector for region r_l is (u^{r_l}, v^{r_l}), the displacement search ranges for region $r_k \in \mathcal{R}_k$ will be defined in a neighborhood around (u^{r_l}, v^{r_l}) as $\left[u^{r_l} - \Delta_k, u^{r_l} + \Delta_k\right]$ and $\left[v^{r_l} - \Delta_k, v^{r_l} + \Delta_k\right]$, each of which is sampled with δ_k to setup the corresponding displacement LUT $\mathcal{D}_{r_k}$ for region $r_k \in \mathcal{R}_k$. By decreasing Δ_k and δ_k level by level, an incremental displacement refinement can be achieved.

By using region-dependent displacement search ranges and incrementally refining each region's displacement search ranges level by level, just using a small number of labels can achieve similar quality of sampling to continuous methods, resulting in better efficiency.

4.3. Initial search range probation

As mentioned in section 4.2, we use an image dimension related upper bound (2) to initialize the displacement search range at the coarsest segmentation level. Since the optical flow directions and magnitude in an image can be arbitrary, we have to use a reasonably large α value to safely capture

the full range. However, due to the limitation of memory and efficiency considerations, an affordable number of labels have to be limited. Therefore for large size images, the sampling interval may not be small enough for accurate matching in the coarsest segmentation level. Then the corresponding errors will be propagated to the next level and cannot be recovered.

To address this issue, we further take advantage of the image pyramid based multiple scale strategy often used in many continuous optical flow methods. In particular, for large images, we first apply our proposed method w.r.t. their half-size version and recover the displacement ranges $[u'_{min}, u'_{max}]$ and $[v'_{min}, v'_{max}]$. Since for smaller search ranges, using the same number of labels enables using a small sampling interval, the result usually contains less errors. Then we apply our proposed method again w.r.t. the original images using $[2u'_{min}, 2u'_{max}]$ and $[2v'_{min}, 2v'_{max}]$ as initial search ranges. If necessary, more pyramid levels can be used. In this paper, we find that a 2- level pyramid is sufficient for our work.

4.4. Energy formulation and optimization

As explained above, our algorithm estimates the optical flow by repeatedly performing discrete energy minimization on multiple-level region trees in a coarse-to-fine way.

Suppose at segmentation level l, the region-tree $\mathcal{T}_l$ in question is defined on region node set $\mathcal{R}_l$ with edge set $\mathcal{E}_l$. Each edge $e_{(i,j)} \in \mathcal{E}_l$ corresponds to a link between two adjacent regions $r_i \in \mathcal{R}_l$ and $r_j \in \mathcal{R}_l$. Each region r_i has $\boldsymbol{N}_{r_i}$ pixels $(x, y) \in r_i$ and is assigned with a label $\mathbb{L}(r_i) \in \mathcal{L}$ after optimization, which corresponds to a 2D displacement vector $(u_{\mathbb{L}(r_i)}, v_{\mathbb{L}(r_i)})$.

Then the data term and smoothness term in (1) are formulated w.r.t. the region-tree labelling $\mathbb{L}(\mathcal{T}_l)$ in a discrete form of

$$E_{data}(\mathbb{L}(\mathcal{T}_l)) = \sum_{r_i \in \mathcal{R}_l} \boldsymbol{\mathcal{M}}(\mathbb{L}(r_i)) \quad (3)$$

and

$$E_{smooth}(\mathbb{L}(\mathcal{T}_l)) = \sum_{e_{(i,j)} \in \mathcal{E}_l} \boldsymbol{\mathcal{S}}(\mathbb{L}(r_i), \mathbb{L}(r_j)) \quad (4)$$

where $\boldsymbol{\mathcal{M}}$ is the matching penalty function evaluating how well the corresponding region $r_i \in \mathcal{R}_l$ is matched between two images $\boldsymbol{I}_t$ and $\boldsymbol{I}_{t+1}$ according to the displacement vector $(u_{\mathbb{L}(r_i)}, v_{\mathbb{L}(r_i)})$ corresponding to label $\mathbb{L}(r_i)$ and $\boldsymbol{\mathcal{S}}$ is the smoothness penalty function evaluating the penalty of assigning two linked regions r_i and r_j with displacement vectors $(u_{\mathbb{L}(r_i)}, v_{\mathbb{L}(r_i)})$ and $(u_{\mathbb{L}(r_j)}, v_{\mathbb{L}(r_j)})$, respectively.

There are many possible definitions for $\boldsymbol{\mathcal{M}}$ and $\boldsymbol{\mathcal{S}}$. In this paper, we define $\boldsymbol{\mathcal{M}}$ based on the well-known zero-mean normalized cross-correlation measure η [25]. In particular, we define

$$\boldsymbol{\mathcal{M}}(\mathbb{L}(r_i)) = \frac{\sum_{(x,y)\in r_i}\left(1.0-\eta(\boldsymbol{I}_t(x,y), \boldsymbol{I}_{t+1}(x+u_{\mathbb{L}(r_i)}, y+v_{\mathbb{L}(r_i)}))\right)}{\boldsymbol{N}_{r_i}} \quad (5)$$

and

$$\boldsymbol{\mathcal{S}}(\mathbb{L}(r_i), \mathbb{L}(r_j)) = |u_{\mathbb{L}(r_i)} - u_{\mathbb{L}(r_j)}| + |v_{\mathbb{L}(r_i)} - v_{\mathbb{L}(r_j)}| \quad (6)$$

That is, the energy function to optimize w.r.t. the region tree $\mathcal{T}_l$ at segmentation level l is

$$E(\mathbb{L}(\mathcal{T}_l)) = \sum_{r_i \in \mathcal{R}_l} \boldsymbol{\mathcal{M}}(\mathbb{L}(r_i)) + \lambda \cdot \sum_{e_{(i,j)} \in \mathcal{E}_l} \boldsymbol{\mathcal{S}}(\mathbb{L}(r_i), \mathbb{L}(r_j)) \quad (7)$$

The tree structure enables the use of efficient DP to optimize (7). In a recursive way, the region tree is bottom-up (leaves-to-root) traversed for label assignment hypothesis evaluation first and then top-down (root-to-leaves) traversed for decision making. For more details, the reader is referred to [18].

Please note that when evaluating a hypothesized label, special attention must be paid in generating the corresponding "*label space image*" (as a generalization of the so-called "*disparity space image*" [21] used in stereo matching) since we are using region-dependent label to displacement mapping and the same label might correspond to different displacement vectors for different regions. Furthermore, for sub-pixel displacements, bilinear image interpolation is performed.

4.5. Post-processing

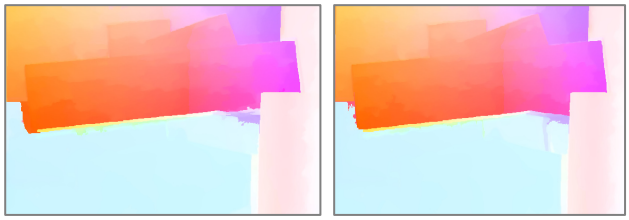
(a) Occlusion detection via cross-checking

The symmetric cross-checking technique is used for correcting optical flow errors due to occlusions. In particular, two optical flows are estimated for images $\boldsymbol{I}_t$ and $\boldsymbol{I}_{t+1}$. Then occlusion reasoning is done by symmetrically cross-checking for consistency violations at the pixel level between these two optical flow fields.

Specifically for each pixel $(x, y) \in \boldsymbol{I}_t$ with recovered optical flow vector (u, v), if its correspondence $(x + u, y + v) \in \boldsymbol{I}_{t+1}$ has optical flow vector of (u', v') and the consistency measure $|u + u'| + |v + v'|$ is greater than a preset threshold β, then pixel (x, y) will be flagged as an inconsistent pixel. Then for each region in the finest segmentation level, if over half of its pixels are flagged as inconsistent, the region will be flagged as occluded.

After all the occluded regions are flagged, a new DP optimization pass is done without the data and smoothness penalties applied to links involving an occluded region so that a larger motion change is made possible. During the bottom-top DP evaluation traversal, an occluded region node will behave as a "pass-through" node, while during the top-bottom DP decision making traversal, an occluded region node will be assigned with its parent node label. This is similar to using neighbor information as done in traditional hole-filling approaches. But the difference is that the chosen neighbor is not found in the spatial domain, but in the region-tree domain in which the parent-child link

is assumed to connect regions with similar attributes based on the region-tree construction procedure. Of course, we have to point out that since our region-tree spans over the whole image, at some points some edges must cross discontinuities, violating such an assumption. Despite its simplicity, this simple approach has shown to give very good performance in all of our experiments.

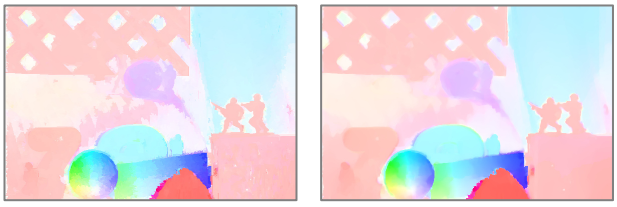


(a) Without cross-checking (b) With cross-checking

Figure 2: Cross-checking based inconsistency detection helps correct errors due occlusions for the "Wooden" dataset, resulting in sharp motion discontinuities.

(b) Continuous optimization based smoothing

Our method can recover very smooth optical flow results by using small region size constraint at the finest segmentation level. However, compared to methods using pixel based representations, there are still noticeable "*graininess*" in some areas since we assume all of the pixels in each region have the same displacement. For better quality, a final local continuous optimization as done in [10] is performed at the pixel level. Since the results from discrete optimization are usually very close to the true displacements, such local optimization mainly acts as a refinement and smoothing step.



(a) Without continuous optimization (b) With continuous optimization

Figure 3: Initialized with optical flow result from discrete optimization, local continuous optimization at the pixel level can smooth out the "graininess" due to the use of translational region motion model. For the "Army" dataset from [1], the final result becomes smoother while the average displacement magnitude change is around 0.05 pixels, which means that results obtained by only using discrete optimization could be quite accurate.

5. Experimental results and evaluation

We use the 2-frame color version of the Middlebury optical flow benchmarking datasets [1] to quantitatively evaluate our proposed method. In particular, 12 image sequences from hidden fluorescent texture, realistic synthetic, stereo and real video categories are tested. In all of the experiments, we used the same set of parameters, which were not explicitly optimized for performance tuning. In particular, we use $M = 2$ level image over-segmentations as it is observed to be enough to provide good performance. The constant $\alpha = 0.04$ is used to initialize the global search range and $\Delta = 0.5$ is used for the local search range refinement. In discrete optimization, 25x25 and 11x11 labels are used for the coarse level and fine level, respectively. The sampling interval δ is correspondingly determined based on the ranges being sampled at each level. As for over-segmentation, the granularity in the coarse level is determined by limiting the region number to be around 3000, while for the fine level, the minimum region size is fixed at 5 pixels. The threshold β is set as 1.0 and the normalized cross-correlation window size is 5x5 for the fine level and 3x3 for the coarse level. As for the smoothness penalty relative weight λ, it is adaptively calculated in the same way as in [18] based on the region-based Kullback-Leiber divergence $\overline{KL}$. That is, $\lambda = \alpha\overline{KL}$ with $\alpha = 0.85$. All these parameters are empirically set and could be further optimized.

In Table 1, we show the average angular error (AAE) and average end-point error (AEPE) of the top four algorithms at the time of submission. Our results of 8 datasets for quantitative evaluation are shown in Figures 2-5. It can be seen that the overall ranking of our method is pretty high (both 4^{th} for AAE and AEPE). In particular, the lowest AAE is obtained for the "Teddy" datasets and the lowest AEPE is obtained for the "Shcefflera" and "Teddy" and "Grove" datasets. One possible reason for relatively inferior AAE performance on other datasets may be due to our current method of discretization. That is, uniformly sampling in the horizontal and vertical displacement search range results in non-uniformity in angular sampling. Moreover, the average performance of our method on the Yosemite sequence also negatively impacts the overall ranking. This could be due to its small image dimension which makes the finest granularity regions still not fine enough to capture subtle motion details. We have found that using different parameters specific to this dataset could improve its performance to some extent. On the other hand, our method obtained superior evaluations around motion discontinuities in most datasets, showing our region-based representation has advantage in preserving motion boundaries. Furthermore, using continuous optimization gives slightly better statistics than the one without using it and boosts the overall ranking by approximately one position. As shown in Figure 4, we compare the performance difference between using multiple level coarse-to-fine region trees and the traditional single-level one as in [18]. Specifically, optical flow result is also obtained as shown in the second column by using only the finer level regions. However, coarse-to-fine displacement refinement is still performed. As we can see, more errors are incurred along motion boundaries compared to the result using two-level region-trees as shown in the third column. For reference, the result from the state-of-art

Table 1: Middlebury benchmark ranking (Average angle & endpoint errors). Red color highlights rows are our method.

Average angle error	avg. rank	Army (Hidden texture) GT im0 im1 all	Army disc	Army untext	Mequon (Hidden texture) GT im0 im1 all	Mequon disc	Mequon untext	Schefflera (Hidden texture) GT im0 im1 all	Schefflera disc	Schefflera untext	Wooden (Hidden texture) GT im0 im1 all	Wooden disc	Wooden untext	Grove (Synthetic) GT im0 im1 all	Grove disc	Grove untext	Urban (Synthetic) GT im0 im1 all	Urban disc	Urban untext	Yosemite (Synthetic) GT im0 im1 all	Yosemite disc	Yosemite untext	Teddy (Stereo) GT im0 im1 all	Teddy disc	Teddy untext
Complementary OF [27]	4.9	4.44 10	11.2 7	4.04 11	2.51 2	9.77 3	**1.74** 1	3.93 4	10.6 4	**2.04** 1	3.87 8	18.8 5	2.19 6	**3.17** 1	**4.00** 1	2.92 4	4.64 7	13.8 3	3.64 6	2.17 5	3.36 3	2.51 13	3.08 2	7.04 2	3.65 9
Adaptive [26]	5.3	**3.29** 1	**9.43** 1	**2.28** 1	3.10 7	11.4 8	2.46 7	6.58 11	15.7 11	2.52 6	**3.14** 1	15.6 2	**1.56** 1	3.67 9	4.46 6	3.48 8	**3.32** 1	13.0 2	**2.38** 1	2.76 12	4.39 12	1.93 8	3.58 4	8.18 4	2.88 3
Aniso. Huber-L1 [28]	6.7	3.71 3	10.1 3	3.08 3	4.36 14	13.0 9	3.77 13	6.92 13	15.3 9	3.60 14	3.54 3	15.9 3	2.04 5	3.38 3	4.45 5	2.47 2	3.88 3	**12.9** 1	2.74 2	3.37 16	4.36 11	2.85 15	3.16 3	7.52 3	2.90 4
DPOF [21]	7.2	5.12 14	12.9 14	3.49 8	3.07 6	10.3 4	2.44 6	**3.09** 1	7.47 2	2.43 5	3.42 2	**12.9** 1	2.41 10	3.55 6	4.56 9	3.35 6	4.69 8	14.2 4	5.14 8	3.59 18	4.67 16	3.83 21	**2.00** 1	**4.93** 1	**1.65** 1

Average end-point error	avg. rank	Army (Hidden texture) GT im0 im1 all	Army disc	Army untext	Mequon (Hidden texture) GT im0 im1 all	Mequon disc	Mequon untext	Schefflera (Hidden texture) GT im0 im1 all	Schefflera disc	Schefflera untext	Wooden (Hidden texture) GT im0 im1 all	Wooden disc	Wooden untext	Grove (Synthetic) GT im0 im1 all	Grove disc	Grove untext	Urban (Synthetic) GT im0 im1 all	Urban disc	Urban untext	Yosemite (Synthetic) GT im0 im1 all	Yosemite disc	Yosemite untext	Teddy (Stereo) GT im0 im1 all	Teddy disc	Teddy untext
Adaptive [26]	5.0	**0.09** 1	**0.26** 1	**0.06** 1	0.23 7	0.78 6	0.18 6	0.54 11	1.19 13	0.21 5	**0.18** 1	0.91 3	**0.10** 1	0.88 4	1.25 4	0.73 6	0.50 2	1.28 2	0.31 2	0.14 11	0.16 13	0.22 10	0.65 3	1.37 3	0.79 4
Complementary OF [27]	6.4	0.11 6	0.28 5	0.10 10	**0.18** 1	0.63 2	**0.12** 1	0.31 4	0.75 4	**0.18** 1	0.19 2	0.97 5	0.12 4	0.97 11	1.31 6	1.00 12	1.78 21	1.73 8	0.87 16	0.11 5	0.12 3	0.22 10	0.68 4	1.48 4	0.95 8
Aniso. Huber-L1 [28]	7.1	0.10 3	0.28 5	0.08 3	0.31 15	0.88 10	0.28 15	0.56 13	1.13 10	0.29 15	0.20 5	0.92 4	0.13 7	0.84 3	1.20 3	0.70 2	**0.39** 1	**1.23** 1	**0.28** 1	0.17 17	0.15 11	0.27 18	0.64 2	1.36 2	0.79 4
DPOF [21]	7.2	0.13 15	0.35 15	0.09 5	0.25 8	0.79 7	0.19 7	**0.24** 1	**0.49** 1	0.21 5	0.19 2	**0.62** 1	0.15 13	**0.74** 1	**1.09** 1	**0.49** 1	0.66 7	1.80 12	0.63 9	0.19 20	0.17 16	0.35 23	**0.50** 1	**1.08** 1	**0.55** 1

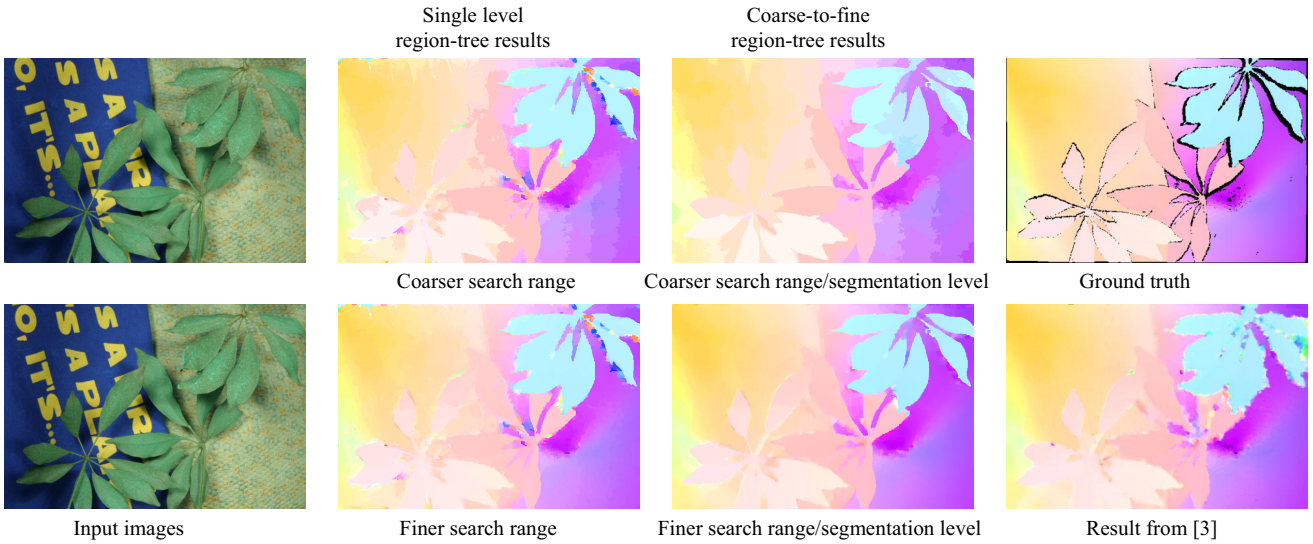


Figure 4: Comparison of using single-level region tree and coarse-to-fine region trees for the "Schefflera" dataset.

continuous optimization based method [3] is also included. By comparison, we can see using over-segmented regions instead of pixels does have unique advantage in handling sharp motion discontinuities.

As for the computation efficiency, take the Urban dataset [1] (image size 640x480 and max displacement is more than 40 pixels) as an example, our un-optimized implementation takes a total running time of about 261 seconds on a PC with a dual-core AMD 2.2GHz CPU.

6. Conclusion and discussion

In this paper, we have presented a new C2F region-tree based method for accurate optical flow estimation using dynamic programming optimization. By using C2F region-tree based image representation and incremental displacement search range refinement, good trade-off between enabling matching primitive to contain enough information through larger support area and reducing the risk of violating the region motion parameterization assumption is achieved. The proposed method can produce sharp motion discontinuities through coarser segmentation while it is also capable of recovering subtle details through finer segmentation. The promising results on the Middlebury benchmarking datasets show the effectiveness of our method.

As for future work, we plan to investigate the use of polar coordinates based parameterization in displacement discretization.

7. References

[1] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, R. Szeliski. A database and evaluation methodology for optical flow. ICCV 2007, http://vision.middlebury.edu/flow/.

[2] B. K. P. Horn, B. G. Schunck. Determining optical flow. Artificial Intelligence., 17(1-3):185-203, 1981.

[3] T. Brox, A. Bruhn, N. Papenberg, J. Weickert. High accuracy optical flow estimation based on a theory for warping. ECCV (4) 2004: 25-36.

[4] M. J. Black and P. Anandan. The robust estimation of multiple motions: parametric and piecewise smooth flow fields. Computer Vision and Image Understanding, 63(1):75–104, January 1996.

Figure 5: Example results (2nd and 4th columns) on the Middlebury datasets along with ground truths (1st and 3rd columns)

[5] I. Cohen. Nonlinear variational method for optical flow computation. In Proc. Eighth Scandinavian Conference on Image Analysis, volume 1, pages 523–530, Tromsø, Norway, May 1993.

[6] A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, and C. Schnörr. Variational optic flow computation in real-time. IEEE Transactions on Image Processing, 14(5):608–615, May 2005.

[7] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. TPAMI, 23(11):1222–1239, 2001.

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. CVPR 2004, v. 1, pp. 261–268.

[9] M. L. Gong, Y. H. Yang. Estimate large motions using the reliability-based motion estimation algorithm. International Journal of Computer Vision 68(3): 319-330 (2006).

[10] V. Lempitsky, S. Roth, and C. Rother. FusionFlow: Discrete-continuous optimization for optical flow estimation. CVPR 2008.

[11] W. Trobin, T. Pock, D. Cremers, and H. Bischof. Continuous energy minimization via repeated binary fusion. ECCV 2008.

[12] B. Glocker, N. Paragios, N. Komodakis. Optical flow estimation with uncertainties through dynamic MRFs. CVPR 2008.

[13] M. J. Black and A. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. IEEE Trans. on Pattern Analysis and Machine Intelligence, 18(10):972–986, October 1996.

[14] C. L. Zitnick, N. Jojic, S. B. Kang. Consistent segmentation for optical flow estimation. ICCV 2005: 1308-1315.

[15] M. Bleyer, C. Rhemann, M. Gelautz: Segmentation-based motion with occlusions using graph-cut optimization. DAGM-Symposium 2006: 465-474

[16] L. Xu, J. Chen, and J. Jia. Segmentation based variational model for accurate optical flow estimation. ECCV 2008.

[17] C. L. Zitnick, S. B. Kang. Stereo for image-based rendering using image over-segmentation. International Journal of Computer Vision 75(1): 49-65 (2007).

[18] C. Lei, J. M. Selzer, Y. H. Yang. Region-tree based stereo using dynamic programming optimization. CVPR (2) 2006: 2378-2385.

[19] C. Stiller and J. Konrad. Estimating motion in image sequences: a tutorial on modeling and computation of 2D motion. IEEE Signal Processing Magazine, 16(4):70-91, 1999.

[20] J. Weickert, A. Bruhn, T. Brox and N. Papenberg. A survey on variational optic flow methods for small displacements, Mathematical Models for Registration and Applications to Medical Imaging (2006), pp. 103-13.

[21] Y. Yang, A. Yuille, and J. Lu. Local, global, and multilevel stereo matching. CVPR 1993, pages 274–279, New York, June 1993.

[22] R. C. Bolles, J. Woodfill. Spatiotemporal consistency checking of passive range data. International Symposium on Robotics Research, 1993.

[23] H.-H. Nagel, W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. TPAMI, Vol. 8, 565-593, 1986.

[24] C. M. Christoudias, B. Georgescu, P. Meer. Synergism in low level vision. ICPR, vol. IV, 150-155, 2002.

[25] O. Faugeras, B. Hotz, H. Mathieu, T. Viille, Z. Zhang, P. Fua, E. Theron, L. Moll, G. Berry, J. Vuillemin, P. Bertin and C. Proy. Real time correlation-based stereo: Algorithm, implementations and applications. INRIA, Tech. Report. RR-2013, 1993.