Sparse Regularization of TV-L¹ Optical Flow

Joel Gibson and Oge Marques

Department of Computer & Electrical Engineering and Computer Science
Florida Atlantic University, Boca Raton, FL, USA
{jgibso19, omarques}@fau.edu

Abstract. Optical flow is an ill-posed underconstrained inverse problem. Many recent approaches use total variation (TV) to constrain the flow solution to satisfy color constancy. We find that learning a 2D overcomplete dictionary from the total variation result and then enforcing a sparse constraint on the flow improves the result. A new technique using partially-overlapping patches accelerates the calculation. This approach is implemented in a coarse-to-fine strategy. Our results show that combining total variation and a sparse constraint from a learned dictionary is more effective than total variation alone.

Keywords: Optical Flow, Sparsity, Dictionary Learning.

1 Introduction

The total variation constraint of the optical flow is remarkably effective for its simplicity. It is however a blunt instrument. Despite its ability to promote smoothness while permitting discontinuities it does not have the capacity to capture additional structure of the flow.

In this work we use total variation to estimate flow. Overlapping patches are created from this flow. We learn an overcomplete dictionary which allows a sparse representation of these patches. By *sparse* we mean that almost all of the vector entries are zero. Intuitively speaking, it means we can represent any flow patch for a given image sequence accurately with only a few patches from the dictionary. An overcomplete dictionary is distinguished from a basis in that it has more elements than the dimensionality it seeks to span. This encourages the representation to be more sparse than if we had just enough elements to span the space.

In our work we show that learning an overcomplete dictionary and combining a sparseness penalty improves flow results over total variation alone. The intent is not to produce the highest ranking algorithm on the Middlebury [1] evaluation site but to take well-known TV algorithms and show that sparse coding improves their result over a suite of test sequences. This is the first work, to our knowledge, that learns an overcomplete dictionary from the flow being computed.

The contributions of this paper are: (i) We demonstrate dictionary learning from the actual sequence instead of learning ground-truth flows in the leave-one-out approach of [11]. (ii) We compute flow patches with less overlap to dramatically improve the computational complexity without sacrificing accuracy.

2 Previous Work

We will look at some highlights in the evolution of optical flow methods that have led to our work. For a recent survey of the field see [4].

Most methods on the Middlebury Flow evaluation site [1] have some roots in Horn and Schunck's seminal paper [10]. Horn and Schunck posed a global description of optical flow in which they minimize the ℓ_2 norm of the color constancy error balanced against flow variation. The variational constraint serves to smooth and propagate the flow over homogeneous areas.

These two fundamental elements of flow estimation remain the foundation of modern methods. Brox et al. [6] made a big improvement to this model by replacing the ℓ_2 norm with ℓ_1 . For the flow this implements total variation, which preserves natural discontinuities while still enforcing smoothness. Furthermore, color constancy constraint is prone to large errors due to noise, lighting change, reflections or occlusions. So additional improvements are seen from using the more computationally challenging ℓ_1 norm on the color constancy term as well.

In an effort to improve the natural inconsistencies in color constancy due to lighting or specular highlights the use of gradient constancy is useful. Using ROF [2,17] structure and texture decomposition on the input image achieves similar results without the overhead of the additional gradient constancy term in the objective.

There has recently been some exploration into using sparse representation as a regularizer of optical flow. This was started with Shen and Wu [15]. They used Haar wavelets as a basis for the flow, enforcing sparse coefficients on a single small-scale image. They worked with quarter-size images from Middlebury and showed improved results over simplistic flow models.

Jia et al. [11] learns overcomplete dictionaries from ground-truth flow fields [4]. They make the computationally-easing assumption that it can be learned with separate horizontal and vertical dictionaries. From their learned dictionary a sparse representation of the flow solution is sought. Their work does show that with a great dictionary, in their case ground-truth-based, sparse solutions are effective regularizers of color constancy. Our work is distinctly different in that we learn an overcomplete dictionary from the flow we are computing. That is, we do not need or use ground truth in our dictionary computation.

3 Our Work

This work was inspired in part by the image denoising success of Elad and Ahron [8] and Mairal et al. [13]. They broke the image into maximum overlapping patches then trained an overcomplete dictionary from those patches. By representing the patches with only a sparse number of dictionary elements the structure of the image was reproduced without the noise. In this sense we want to capture flow structure in a dictionary and use a sparse representation of these flow patches to add regularization to traditional flow estimation. However, differences quickly emerge since the flow we want to learn is not given as the image was in the denoising template. This implies there must be some bootstrap step to construct a flow approximation which can be learned and then iteratively refined.

3.1 Partially-Overlapping Patches

Looking for sparse representations of patches stems from the reality that image-sized signals are too large to look for sparse representations from a computational tractability standpoint. The assumption then is that by studying patches we will find that the structure of the signal will be some sparsely representable lower dimensional signal embedded in the higher dimension of the image space. The computational complexity of sparse learning and representation is a combinatorial function of the patch size, dictionary size, and number of patches. So we are motivated to find the smallest patches and smallest number of patches that effectively capture the structure of the flow.

Given an image with P patches of size n , a $\beta \times$ overcomplete dictionary $D \in \mathbb{R}^{2n \times 2\beta n}$ and a flow $v \in \mathbb{R}^{2N \times 1}$, we want to find a sparse representation $a_{ij} \in \mathbb{R}^{2\beta n \times 1}$ for each patch. We can express this as

$$\hat{a}_{ij} = \arg \min_{a_{ij}} \mu \|a_{ij}\|_0 + \|Da_{ij} - R_{ij}v\|_2^2 \quad (1)$$

where $R_{ij} \in \{0, 1\}^{2n \times 2N}$ extracts the ij th patch from the flow and $\mu \in \mathbb{R}$ is a penalty weight of sparsity versus data fidelity. The patch and image have n and N elements respectively. This problem is NP-hard but approximate solutions exist. We use a maskable Orthogonal Matching Pursuit (OMP) algorithm [12].

In the denoising work of [8,13] image patches are offset by one pixel horizontally and vertically. Reconstructing the image consists of averaging these patches together, thereby eliminating patch-sized blockiness. This is straightforward to implement but creates a large albeit sparse representation. Here the number of patches P is approximately the same the number of pixels N . In our work we consider 10×10 patches with a $4 \times$ overcomplete dictionary. In this case the sparse representation a would be approximately 800 times larger than the original image. This becomes intractable for all but tiny images. We have observed useful results with cardinalities of 2–15, still producing quite large sparse structures.

The motivation for using partially overlapping patches stems from asking whether we can reduce the number of patches and hence complexity while not sacrificing accuracy. The number of patches is approximately proportional to $1/k^2$ where k is the offset. There is clearly a large computational advantage to using values of $k > 1$.

As an experiment to evaluate the error introduced from partial overlap we learn a set of overcomplete dictionaries with 10×10 patches from each of the eight ground truth flows from Middlebury [1]. Then for each test sequence we create patches with offset $k = 1, \dots, 10$. We reconstruct the flow from the patches and measure the Average Endpoint Error (Avg EE) [4]. We found the average of all of these Average Endpoint Errors is very close to linear with respect to the offset. This suggests, as experiments later show, that partial-overlapping patches can reduce the computational complexity of sparse representation by an order of magnitude with minimal accuracy impact.

3.2 Dictionary Learning

Consider the same variables used in Eq. 1 except now we want to create an ideal dictionary for sparse representation of the flow. This problem can be written as

$$\hat{D} = \arg \min_{D, a_{ij}} \sum_{ij} \|a_{ij}\|_0 \quad \text{subject to} \quad \|Da_{ij} - R_{ij}v\|_2^2 < \epsilon. \quad (2)$$

We find an approximate solution to this NP-hard problem using a Method of Optimal Directions (MOD) algorithm first described by Engan, et al. [9]. This iterative method discards the globally least used atom and adds an atom in the direction of the largest error. We modified an implementation by Elad [7] to allow data masking.

Fig. 1. 2D learned flow dictionary for Rubber Whale, 8×8 patches, $2 \times$ overcomplete

Figure 1 shows an example of a flow dictionary. Since the elements themselves are 2D vectors they are color-coded in the Middlebury tradition.

3.3 Sparse Total Variation

A typical optical flow objective function using a total variation regularizer looks like

$$\arg \min_v \{ \|\rho(v)\|_p + \lambda_{TV} (\|Gv_1\|_{1,2} + \|Gv_2\|_{1,2}) \} \quad (3)$$

where $\rho(v) = \nabla I \cdot v + I_t$, I is the image intensity, $v = (v_1, v_2)$ is the optical flow, I_t is the image temporal derivative, $G = (\partial_x, \partial_y)^T$, and λ_{TV} is the relative weighting of total variation penalty. The first term represents the color constancy penalty with an ℓ_p norm. As proposed by Brox et al. [6] and Wedel et al. [17], we consider $p = 1$ since the ℓ_1 norm more closely matches the long-tailed color constancy error.

We also consider another version from Werlberger et al. [18] where $G = D^{1/2} \nabla$ and $D^{1/2}$ is the diffusion tensor. This allows the flow crossing orthogonal to an image edge to change without penalty while smoothly increasing the penalty as the direction changes to parallel the image edge.

Algorithm 1. Sparse Total Variation**Input:** We are given input images $I(x, t_0), I(x, t_1)$, patch size n , and offset k **Output:** Optical flow v

```

1 Construct pyramid of images  $I^j$  for  $j = 1, \dots, \text{pyramidLevels}$ ;
2  $v \leftarrow 0$ ;
3 for  $j \leftarrow \text{pyramidLevels}$  down to 1 do
4   for  $i \leftarrow 1$  to  $TV\text{warps}$  do
5      $v \leftarrow \text{Nesterov}(\nabla I, I_t, v)$  Eq. 3;
6     Warp  $I^j(x, t_1)$ , calculate  $I_t$ ;
7   end
8    $D^i \leftarrow \text{OvercompleteDCT}$ ;
9   for  $i \leftarrow TV\text{warps} + 1$  to  $\text{warps}$  do
10     $\text{flowPatches} \leftarrow \text{FlowToColumn}(v, n, k)$ ;
11     $D^{i+1} \leftarrow \text{MOD}(\text{flowPatches}, D^i)$  Eq. 2;
12     $a \leftarrow \text{OMP}(v, D^{i+1})$ ;
13    for  $l$  to  $\text{iterations}$  do
14       $v \leftarrow \text{Nesterov}(\nabla I, I_t, a, v, D^i)$  Eq. 5;
15       $a \leftarrow \text{OMP}(v, D^i)$  Eq. 1;
16    end
17    Warp  $I^j(x, t_1)$ , calculate  $I_t$ ;
18  end
19 end
20 Return  $v$ ;
```

We add to Eq. 3 a sparsity penalty and a coupling term between the sparse representation and the flow.

$$\arg \min_{v, a_{ij}} \left\{ \|\rho(v)\|_1 + \lambda_{SP}(\|Gv_1\|_{1,2} + \|Gv_2\|_{1,2}) + \mu \sum_{ij} \|a_{ij}\|_0 + \frac{\tau}{2} \sum_{ij} \|R_{ij}v - Da_{ij}\|_2^2 \right\} \quad (4)$$

To make Eq. 4 easier to approximate we break it into two subproblems. First we assume that a is fixed which yields a convex problem in v . Secondly we fix v and solve for a . We iterate between these two subproblems a few times.

The first subproblem is described by

$$\arg \min_v \left\{ \|\rho(v)\|_1 + \lambda_{SP}(\|Gv_1\|_{1,2} + \|Gv_2\|_{1,2}) + \frac{\tau}{2} \sum_{ij} \|R_{ij}v - Da_{ij}\|_2^2 \right\}. \quad (5)$$

We solve (a smoothed approximation of) this problem using Nesterov's method [14] similar to the application in [3,5]. For a given smoothing parameter α , the Lipschitz constant is

$$L = \frac{8\lambda}{\alpha} + \frac{\|\nabla I^T \nabla I\|_2}{\alpha} + \tau \|R^T R\|_2. \quad (6)$$

This algorithm allows an optimal convergence rate that scales with $\mathcal{O}(1/k^2)$ where k is the number of iterations even for this non-smooth objective. The second subproblem is solved independently for each patch using OMP and has already been described in Eq. 1. The individual pieces are shown assembled in Algorithm 1.

4 Experimental Results

Experimental results are presented for four different dictionary types described below and two versions of the TV optical flow. Lastly, the average results of the tensor-directed version is shown for different patch offsets.

4.1 Dictionaries

Experiments were conducted using the following dictionaries:

- DCT: A predefined overcomplete DCT dictionary is used. This is the only instance where the dictionary is not learned and serves as a comparison for the effectiveness of a learned dictionary.
- 1D: Optical flow is a two dimensional vector. Patches of the horizontal and vertical elements of the flow are concatenated and a single dictionary is learned.
- 2×1D: Here two separate dictionaries are learned: one from the horizontal flow patches, the other from the vertical patches. This is the dictionary method used by Jia et al. [11].
- 2D: For 2D the horizontal and vertical patches are combined at each location, into a single patch of $2n$ elements. A single dictionary is then learned in this higher dimension.

The complexity and speed of computation is related to the number of dictionary elements. For that reason it seems meaningful to compare dictionaries of similar size. So the $4\times$ overcomplete 2D dictionary is compared to the $8\times$ overcomplete 1D dictionary because they both have the same number of dictionary elements.

4.2 Implementations Details

In the experiments shown for Middlebury images, there are 5 pyramid levels, 5 bootstrap warps and 10 warps with sparsity. We iterate 4 times in innermost loop of Algorithm 1.

These are the constants used in Eqs. 1, 2, 3, and 4. The OMP weighting constant $\mu = 0.05$ was used for all cases. Ten iterations of the MOD dictionary learning were run on each invocation with $\epsilon = 0.001$. For the ℓ_1 proximal function, $\alpha = 0.01$. Following [3], the $\lambda_{TV} = 0.0006$ on the coarsest level and increases to 0.05 at the finest level. Similarly $\lambda_{SP} = 0.005$ and $\tau = 0.02$ on the finer levels increasing to 0.02 and 0.16 respectively at the top level.

A standard ROF decomposition of structure and texture was performed on the input of all experiments and a 5×5 median filter was applied after each flow calculation as recommended by Sun et al. [16].

Table 1. Tensor-directed and TV-L¹ Average Endpoint Error (Avg EE) for Middlebury training set. Patch size is 10×10

Image Dict. Type Overcomplete Dictionary Size	Tensor-Directed					TV-L ¹				
	Baseline	DCT	1D	2×1D	2D	Baseline	DCT	1D	2×1D	2D
		4× 400	4× 400	2× 400	2× 400		8× 800	8× 800	4× 800	4× 800
Dimetrodon	0.1576	0.1626	0.1524	0.1516	0.1512	0.1562	0.1810	0.1782	0.1777	0.1766
Grove2	0.2000	0.2025	0.1766	0.1781	0.1783	0.1921	0.2003	0.1743	0.1727	0.1775
Grove3	0.6630	0.7415	0.6808	0.6868	0.6896	0.6869	0.7199	0.6489	0.6520	0.6504
Hydrangea	0.1633	0.1678	0.1596	0.1595	0.1620	0.1597	0.1640	0.1572	0.1569	0.1568
RubberWhale	0.1169	0.1171	0.1119	0.1120	0.1103	0.1118	0.1020	0.0993	0.0984	0.0985
Urban2	0.3761	0.4806	0.3540	0.3531	0.3427	0.3801	0.5560	0.3440	0.3415	0.3430
Urban3	0.6051	0.6776	0.5711	0.5737	0.5654	0.6441	0.6425	0.5382	0.5462	0.5405
Venus	0.3035	0.3599	0.2831	0.2815	0.2818	0.2977	0.3587	0.2733	0.2736	0.2715
Avg % improved	0	-9.69	4.63	4.53	4.99	0	-10.67	5.95	6.08	6.03

In order to form a whole set of patches with offset k from an image it is necessary that $[h \ w] - [m \ n] = 0 \bmod k$, where h, w and m, n are the image and patch dimensions respectively. It may be necessary to augment the image by a small apron to maintain this relationship. The apron pixels are labeled with an out-of-bounds marker similar to the Middlebury ground truth occluded pixels. These values are masked off during the sparse learning and representation steps.

4.3 Discussion

The DCT dictionary performs worse than the baseline of no sparsity. It was seen in experiments described at the end of Section 3.1, that sparse representations of the flow can be obtained with the DCT. While a sparse representation is a necessary condition for improvement, the DCT does not apparently capture the structure of the flow in a meaningful way. On the other hand we see that on the average, every learned dictionary method presented in Table 1 outperforms the baseline.

We found 10×10 patches outperformed smaller patch sizes suggesting that the larger patch size is more effective at capturing the flow structure.

We computed and compared the tensor-directed percentage Avg EE change relative to baseline averages for all possible offsets with 10×10 patches. Increasing the patch offset actually decreased the Avg EE error up to $k = 6$. For small patch offsets it was observed that averaging more patch approximations together creates both a smoother flow with more rounded edges, even where sharp angles are desired. For flow with large smooth areas, small patch offset will smooth the flow better than increasing the TV penalty. However, the errors incurred by rounding corners and edges often create a greater loss.

This algorithm may perform worse than baseline when at a coarser level the TV-only bootstrap flow makes a poor choice in an ambiguous area. This structure is later sometimes learned into the dictionary and then it is encouraged to persist by the sparseness penalty. In most but not all cases this is self-rectifying.

5 Conclusions

We have shown that flow structure can be learned from actual sequences in a bootstrap manner and used to further refine flow computation. Overcomplete dictionary learning and sparse flow representation have been demonstrated with generic total variation algorithms. The proposed method could easily be added to any more sophisticated variational approach. We have introduced a method of partial-overlapping patches that offers dramatic acceleration in the computation of this sparse representation. This methodology should also be useful in other patch-based applications such as denoising.

References

1. The Middlebury Computer Vision Pages (2013), <http://vision.middlebury.edu>
2. Aujol, J., Gilboa, G., Chan, T., Osher, S.: Structure-texture image decomposition modeling, algorithms, and parameter selection. *IJCV* 67(1), 111–136 (2006)
3. Ayvaci, A., Raptis, M., Soatto, S.: Sparse Occlusion Detection with Optical Flow. *IJCV* 97(3), 322–338 (2011)
4. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A Database and Evaluation Methodology for Optical Flow. *IJCV* 92(1), 1–31 (2011)
5. Becker, S., Bobin, J., Candès, E.: NESTA: A fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences* 91125, 1–37 (2011)
6. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Pajdla, T., Matas, J. (eds.) *ECCV 2004*. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004)
7. Elad, M.: *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, 1st edn. Springer Publishing Company (2010) (incorporated)
8. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing* 15(12), 3736–3745 (2006)
9. Engan, K., Aase, S.O., Hakon Husoy, J.: Method of optimal directions for frame design. In: *Proceedings of the Acoustics, Speech, and Signal Processing*, pp. 2443–2446 (1999)
10. Horn, B., Schunck, B.: Determining optical flow. *Artificial Intelligence* 17, 185–203 (1981)
11. Jia, K., Wang, X., Tang, X.: Optical Flow Estimation Using Learned Sparse Model. In: *Proceedings of IEEE International Conference on Computer Vision*. No. 60903115 (2011)
12. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online learning for matrix factorization and sparse coding. *The Journal of Machine Learning Research* 11, 19–60 (2010)
13. Mairal, J., Elad, M., Sapiro, G.: Sparse representation for color image restoration. *IEEE Transactions on Image Processing* 17(1), 53–69 (2008)
14. Nesterov, Y.: Smooth minimization of non-smooth functions. *Mathematical Programming* 103, 127–152 (2005)
15. Shen, X., Wu, Y.: Sparsity model for robust optical flow estimation at motion discontinuities. In: *Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 2456–2463. IEEE (2010)
16. Sun, D., Roth, S., Black, M.: Secrets of optical flow estimation and their principles. In: *Computer Vision and Pattern Recognition (CVPR)*, pp. 2432–2439. IEEE (2010)
17. Wedel, A., Pock, T., Zach, C., Bischof, H., Cremers, D.: An Improved Algorithm for TV-L1 Optical Flow. In: Cremers, D., Rosenhahn, B., Yuille, A.L., Schmidt, F.R. (eds.) *Statistical and Geometrical Approaches to Visual Motion Analysis*. LNCS, vol. 5604, pp. 23–45. Springer, Heidelberg (2009)
18. Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., Bischof, H.: Anisotropic Huber-L1 optical flow. In: *Proceedings of BMVC* (2009)