Optical Flow Estimation with Channel Constancy

Laura Sevilla-Lara[1], Deqing Sun[2], Erik G. Learned-Miller[1], and Michael J. Black[3]

[1]School of Computer Science, University of Massachusetts Amherst, MA, USA
{lsevilla, elm}@cs.umass.edu
[2]School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA
dqsun@seas.harvard.edu
[3]Max Planck Institute for Intelligent Systems, Tübingen, Germany
black@tuebingen.mpg.de

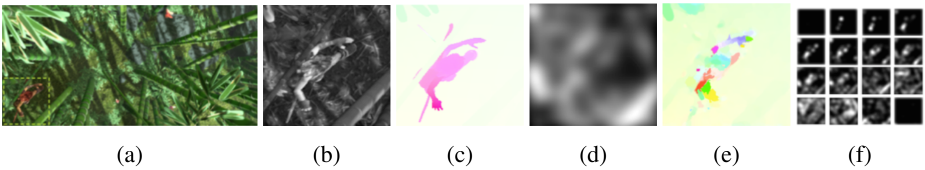
Abstract. Large motions remain a challenge for current optical flow algorithms. Traditionally, large motions are addressed using multi-resolution representations like Gaussian pyramids. To deal with large displacements, many pyramid levels are needed and, if an object is small, it may be invisible at the highest levels. To address this we decompose images using a *channel representation* (CR) and replace the standard brightness constancy assumption with a descriptor constancy assumption. CRs can be seen as an over-segmentation of the scene into layers based on some image feature. If the appearance of a foreground object differs from the background then its descriptor will be different and they will be represented in different layers. We create a pyramid by smoothing these layers, without mixing foreground and background or losing small objects. Our method estimates more accurate flow than the baseline on the MPI-Sintel benchmark, especially for fast motions and near motion boundaries.

Keywords: Optical flow, channel representation, pyramids, large motions

1 Introduction

Small, fast moving objects are easy for humans to see and track. Their motion is important for biological tasks such as obstacle avoidance, catching, and predator detection. Figure 1(a) shows an example in which a small animated character is viewed from above, running through a bamboo forest [7]. In contrast to biological vision, current optical flow algorithms perform badly in such cases (Fig. 1(e)). We find that this is particularly true for small or thin regions.

The issue stems from the basic assumptions of most current flow methods. Most techniques estimate dense optical flow using two constraints: brightness constancy and spatial smoothness of the flow field [28]. Brightness constancy assumes that the intensity value of a small region remains constant despite its change in location. The brightness term is a non-linear function of the flow and, in gradient-based formulations, is typically linearized for optimization. This linearization is valid only in the case that the displacement is small. In order to capture longer range motion, a coarse-to-fine method is employed [3,5], typically using a Gaussian pyramid [6]. The pyramid is built by successively smoothing and downsampling the images. The problem with this approach is



(a) (b) (c) (d) (e) (f)

Fig. 1: **Problems with Gaussian pyramids:** (a) Image from [7] (b) Detail of a small, fast object (c) Ground truth (d) Blurred image patch from high pyramid level (e) Flow using a Gaussian pyramid [27] (f) CR of the same patch blurred with same kernel size (g) Flow using a CR pyramid and our method.

that, for scenes with multiple moving objects, this blurs the pixel values across object boundaries. For small or thin objects this means that at coarse (high) levels of the pyramid, the object may completely disappear; see Fig. 1(d-e) for an example of a blurred image region and the flow of the Classic+NL algorithm, which uses a pyramid [27].

Instead, if one could segment the scene into objects, then the objects could be matched across large displacements. But since object segmentation is itself an unsolved problem, we need an alternative. In this work we replace the brightness constancy assumption with a descriptor constancy assumption. For this we represent an image using a channel representation (CR). This representation contains a descriptor at each pixel location. This descriptor is a locally weighted histogram.

The advantage of this representation is that performing blurring in CR space does not introduce mixing of the brightness values of the pixels. Instead, the image is decomposed into several different channels, according to the pixel intensities (or other image property). Each of these channels is then blurred separately (Fig. 1(f)). This process allows spreading the information about the pixel values spatially. This smooths the optimization landscape, but prevents the averaging of pixel values that one sees in a Gaussian pyramid. An example of the effect of blurring using CRs in the optimization landscape is shown in Fig. 2. This descriptor constancy assumption is also linearized, to fit in the traditional approach of flow estimation. We then apply the standard coarse-to-fine framework, creating a CR-pyramid by blurring each CR and downsampling it. This prevents losing some small objects at the coarse levels and oversmoothing motion boundaries.

In this paper we start with the original Classic+NL technique, and simply replace the brightness-constancy data term by our descriptor-constancy data term, leaving the rest of the system, including the parameter values, untouched. We call this data term Channel constancy or descriptor constancy. The only parameter that we change is the weight of the smoothness term, since the statistics of the values of an image and a CR are slightly different. We compare the performance of both systems in a synthetic setting as well as on the standard benchmark for large displacements, which is the *MPI-Sintel* dataset [7]. We find that this simple change improves results overall, especially in long range motions and at motion boundaries. This suggests that replacing brightness constancy with channel constancy may also improve other optical flow algorithms.

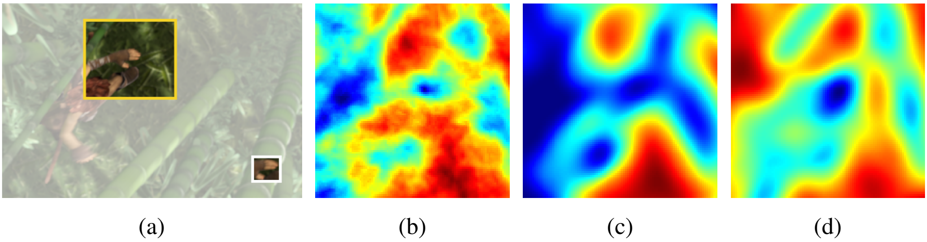


Fig. 2: **Pyramid versus CR pyramid.** Computing the sum of squared difference between the hand in the white region in (a) and pixels in the yellow region gives the error surface in (b) with many local minima. The global optimum is near the center. Using a Gaussian pyramid oversmooths the energy (c) and the minimum is less clear. Decomposing the image into a CR and blurring each channel using the same kernel as in (c) yields the energy in (d), which smooths the surface but maintains the global optimum. The error increases from cold (blue) to warm (red) color.

2 Previous Work

The problem of recovering long range motion has previously been addressed [4,25,31], but results on the *MPI-Sintel* dataset [7] show that current methods still fail to capture really large motions, especially of small objects.

Many of the top performing algorithms on the standard datasets are based on the traditional approach of Horn and Schunck [14]. This method minimizes an energy function that is the sum of two terms: the smoothness term that encourages neighboring flow vectors to be similar, and the data term that encourages corresponding pixels to have the same brightness. These classical methods often use a coarse-to-fine approach [6] for computing optical flow. These methods smooth the images and, as a consequence, the optimization landscape, so that motions larger than 1 pixel can be estimated. However, smoothing the images enough to capture large motions makes it nearly impossible to recover the motion of small objects with large displacements.

In principle, this approach fails because each pixel is not discriminative enough. If we consider estimating optical flow as finding correspondences between image pixels, we would like each pixel to be uniquely identifiable so that the correspondence can be found easily. In the traditional approach, these correspondences cannot be found because the blurring makes pixels indistinguishable from each other. One way of avoiding this is adding additional features at each pixel. Estimating the flow becomes finding correspondences where not only the pixel value matches, but also other pixel features like linear filter responses, edges and information about the neighboring pixels [18,25,26,30,31]. This is typically done by including additional feature matching terms in the original energy function.

However, including these terms in the classical approach is not trivial, because it makes the optimization of the energy function difficult. For these terms to be integrated in the framework, they need to be differentiable and increase away from the global optimum, which is often not the case. Therefore, this family of methods that try to incorporate additional features are difficult to optimize, and suffer from increased computational cost, or in some cases they restrict the solutions to a discretized space that does not reach sub-pixel accuracy.

Brox and Malik [4] take an important step towards incorporating additional features in the energy function. They precompute a descriptor at each pixel, and find the best match at the next frame for each pixel. Then they include a term in the energy function that encourages the estimated flow to be similar to the precomputed best feature match. While this is a step in the right direction, it does not directly include the similarity between descriptors in the global optimization. Our method fully integrates the descriptor matching in the global optimization of the energy function.

The descriptor used in this work and its variants have been successfully applied to a variety of problems in the vision literature [2,17,20,21,24,29]. Representing an image using this descriptor at each pixel has received several names, but here we use the most common: channel representation. The descriptor has been extensively used for image denoising [9,10,11,12,16] and pose estimation [15]. It has also been used for affine image alignment [19]. In object tracking, Sevilla-Lara and Learned-Miller [23] use a distribution over grayscale values at each pixel to create an object template that can be smoothed, to reach long displacements. The success with tracking suggests that the descriptor may also be useful for optical flow. No previous work has considered this and we show that the channel representation helps recover the motion of small, fast moving objects.

3 Methods

In this section we first explain the proposed energy function and its relationship to the traditional approach. We then describe how to compute the image descriptor used in this energy function.

3.1 Energy Function

Most optical flow formulations make assumptions about brightness constancy and spatial smoothness, in one form or another. In this formulation, we minimize the following energy function

$$E(u,v) = E_{\text{brightness}}(u,v) + \lambda E_{\text{smooth}}(u,v), \tag{1}$$

where u and v represent the horizontal and vertical flow fields from the first image I_1 to the second image I_2 respectively.

The first (brightness) term assumes that the brightness of a pixel persists over time and is typically formulated as

$$E_{\text{brightness}}(u,v) = \sum_{x,y} \rho\left(I_1(x,y) - I_2(x + u_{(x,y)}, y + v_{(x,y)})\right). \tag{2}$$

where $u_{(x,y)}$ and $v_{(x,y)}$ represent the flow at a pixel (x,y) and where $\rho(\cdot)$ is a robust penalty function that downweights the influence of outliers (i.e. violations of the brightness constancy assumption) [3].

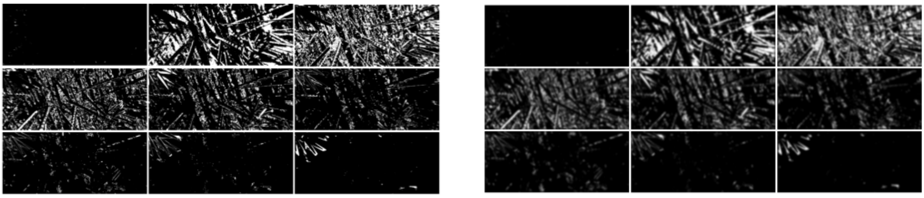


Fig. 3: **Left**: CR after exploding one of the images from *MPI-Sintel* (see Fig. 1). The number of brightness levels (or layers) has been quantized to 9. **Right**: The same CR after smoothing the layers spatially.

The brightness constancy term is a non-linear function of the unknown flow fields. The traditional approach linearizes the brightness constancy term for optimization and requires the smoothing of the optimization landscape. This is typically implemented by blurring the image as part of a coarse-to-fine strategy. Depending on the size of certain objects in the scene and the amount of blur, some details may be lost. To preserve the details, we need to robustly smooth the object boundaries without mixing pixel values. Next we describe the channel representation designed for this purpose.

3.2 Channel Representation

The channel representation (CR) allows robust smoothing over object boundaries. It contains a probability distribution over the feature space at each pixel location. For example, if the image is in grayscale, the probability distribution will be over values from 0 to 255. A CR is built by "exploding" an image, which places a Kronecker delta function at each pixel, according to Eq 3. At each pixel, this yields a distribution that is 0 for every value, except for the pixel's intensity:

$$d(i,j,k) = \begin{cases} 1 & \text{if } \lfloor \frac{I(i,j)}{\triangle} \rfloor = k, \\ 0 & \text{otherwise}, \end{cases} \tag{3}$$

where $\triangle$ is the quantization bin size, variable k ranges from $1 \leq k \leq K$, K is the number of bins used for quantizing the feature space, and $\lfloor \cdot \rfloor$ denotes the floor operation.

We can blur a CR to smooth the optimization landscape, by blurring each of the k channels separately as

$$d_s(k) = d(k) * h_{\sigma_s}, \tag{4}$$

where h is a 2D Gaussian kernel of standard deviation σ_s, in each dimension and $*$ is the convolution operator.

Figure 3 shows a visualization of the result of exploding an image and blurring its resulting CR. Because the layers are blurred separately, pixel values are not mixed across object boundaries. Hence we can preserve the fine image details, which is the main advantage of using CRs. Smoothing maintains the property that there is a distribution at each pixel location.

Pixel intensities may change a little over time, for example due to subpixel motion or changes in illumination. Such situations can be represented by uncertainty in the feature space. We achieve such uncertainty by smoothing in the k direction at each pixel

$$d_{ss}(i,j) = d_s(i,j) * h_{\sigma_f}, \tag{5}$$

where h_{σ_f} is a 1D Gaussian kernel of standard deviation σ_f.

3.3 Channel Constancy Assumption

In our approach, the data term in the energy function enforces descriptor constancy, which matches the descriptor at each pixel instead of the pixel intensity. The energy function becomes

$$E(u,v) = E_{CR}(u,v) + \lambda E_{\text{smooth}}(u,v), \tag{6}$$

where the descriptor constancy term enforces that each of the components in the descriptors should match. Let the descriptor at each pixel have K components, and $d_1(x,y,k)$ be the kth component (level) of the descriptor at pixel (x,y) in image I_1. Then the data term becomes

$$E_{\text{CR}}(u,v) = \sum_{x,y}\sum_{k=1}^{K} \rho\left(d_1(x,y,k) - d_2(x+u_{(x,y)}, y+v_{(x,y)}, k)\right). \tag{7}$$

Two CRs can be compared by comparing each of their corresponding components, as in Eq. 7. In this work we use two different metrics: the $L2$ distance ($\rho(x) = x^2$) and the generalized Charbonnier ($\rho(x) = (x^2+\epsilon^2)^\alpha$) [8]. These choices were made based on previous studies [27] that illustrate their advantages. Note that we apply the function to each component of the CRs, not to the pixel values.

To understand what it means to compute the distance between two CRs, we can consider a simple case in which each input image has one non-zero pixel. We then consider the distance as a function of the difference in pixel values. The CRs blur the pixel value across several levels and the error is computed by summing the robust error across all K levels in Eq. 7. In Fig. 4 we plot this error as a function of pixel-value difference and compare the CR case with the standard intensity case for both the quadratic and Charbonnier penalty functions. The effective shape of the error function is quite different in the case of the CR error. In fact, the error function has an interesting property of saturating like a robust error function. This saturation happens even when the penalty function is quadratic.

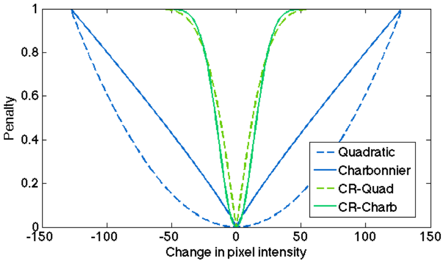


Fig. 4: **Shape of the different penalty functions:** The generalized Charbonnier and quadratic distance functions change shape when applied to a pair of distributions instead of to a pair of pixels. This new shape is more robust to outliers but it is still convex in CR space like the quadratic function.

The intuition here is simple. If there is no blurring across layers, then the difference between two CRs will be zero only when the input pixels have the same value and will be constant for any difference in pixel values; that is, like an inverted delta function. Blur across layers allows different pixel values to be compared and the more smoothing there is the wider the convex region around zero.

4 Optimization

Our goal is to isolate and evaluate the effect of channel constancy versus brightness constancy. To that end we use the existing Classic+NL framework for flow estimation and simply replace the data term while keeping all other elements of the method the same.

4.1 Integration in Traditional Approach

Optimization in Classic+NL involves linearizing the brightness term as part of an incremental warping strategy that gradually warps the second image towards the first image.

By replacing the images with the CR, each incremental warping step linearizes the descriptor constancy term of the energy function (Eq. 7) around the current flow estimate (u_0, v_0), differentiates Eq. 6 w.r.t. the flow increment (du, dv), and sets the derivatives to be zero. The resultant linear equation system to solve for the flow increment are

$$\begin{bmatrix} \sum_k \tilde{\rho}(d) d_x(k)^2 + \lambda L & \sum_k \tilde{\rho}(d) d_x(k) d_y(k) \\ \sum_k \tilde{\rho}(d) d_x(k) d_y(k) & \sum_k \tilde{\rho}(d) d_y(k)^2 + \lambda L \end{bmatrix} \begin{bmatrix} du \\ dv \end{bmatrix} = - \begin{bmatrix} \sum_k d_x(k) d_t(k) + \lambda L u_0 \\ \sum_k d_y(k) d_t(k) + \lambda L v_0 \end{bmatrix},$$

where the weighting function $\tilde{\rho}(x) = 2(x^2 + \epsilon^2)^{a-1}$ for the generalized Charbonnier penalty $\rho(x) = (x^2 + \epsilon^2)^a$, L is the Laplacian operator, d_x and d_y are derivatives of the CR w.r.t. x and y and d_t is the derivative of the CR w.r.t. t. We compute the derivatives by taking the derivative of each layer of the CR, as if they were images.

4.2 Optical Flow Estimation

We use a coarse-to-fine warping-based approach to optimize the proposed energy function. Instead of using the traditional image pyramid [5], we use a pyramid of CRs, as shown in Fig. 5. The first level is computed by exploding the image according to Eq. 3. Each successive level is computed from the previous one by smoothing it as in Eq. 4 and downsampling. We downsample each layer of a CR as if the layer were an image and interpolate using bicubic interpolation.

We compute the flow at the coarsest level of the two pyramids and use the flow to warp the second CR toward the first CR. Each layer of a CR is warped separately. At the next level, the computation of the flow starts from the position of the previous level, interpolating for the points where there is no estimation yet. We use a 3-stage graduated non-convexity (GNC) scheme for the optimization [28]. The first one uses a quadratic penalty, the last one uses a generalized Charbonnier penalty and the middle one uses a linear combination of the two (as in [27]). In practice, since the CRs need to be differentiated, the bottom level of the pyramid is also smoothed with a small Gaussian filter. We call the resulting algorithm *Channel Flow (CFlow)*.

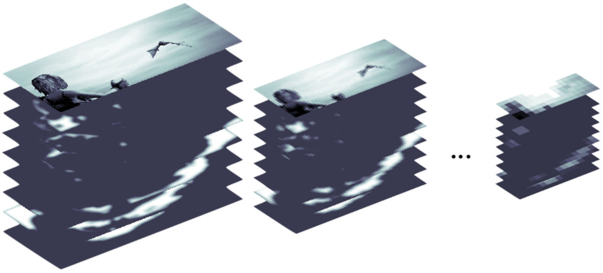


Fig. 5: **Channel representation pyramid.** Each level of the pyramid is a CR, created by smoothing and downsampling the previous level. The original image is also shown here but is not part of the CR.

4.3 Modeling the Change in Illumination

If there were no smoothing across layers in the k (vertical) direction, the CR would be sensitive to slight changes in image intensity. Despite smoothing across layers, we find that the descriptor constancy is still sensitive to brightness variations. Classic+NL does not actually use brightness constancy but rather uses a texture decomposition for the data term, which reduces the effects of illumination change. In the case of the CR, we take a different approach and explicitly model illumination change.

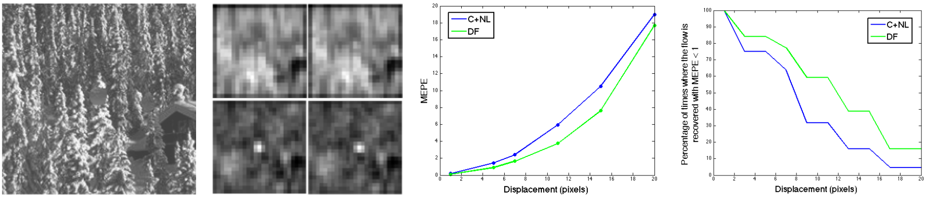
Previous work [13,32] has shown the advantage of using a model of the change in illumination for techniques that assume brightness constancy. The physics of the natural world make the changes in illumination in a scene multiplicative. However, many images have gamma-correction applied to them, which makes the changes in illumination have an additive effect on the image. Therefore, a plausible model for changes in illumination is: $I(x + u_{(x,y)}, y + v_{(x,y)}, t + 1) = I(x, y, t) + b(x, y)$.

What we need, however, is the effect of brightness changes on the CR. If a pixel changes brightness, this changes the *level* at which the pixel appears in the CR. Thus, to compensate for the brightness change we want to *warp* the CR in the direction that undoes this change. This warping is analogous to the warping we do in space using the optical flow except it happens in the vertical direction of the CR.

To better understand how brightness varies, we used the Sintel training sequences. We warped adjacent frames together using the ground truth flow and computed the brightness difference b at every pixel. We found that the distribution of b values is tightly peaked at zero with heavy tails.

This leads us to a simple method to compute the brightness change. Given the current flow estimate, we warp the input images and compute their difference. We then apply a median filter of 21×21 to obtain a robust estimate of b at every pixel. We can use the flow fields and the brightness change to apply a 3D warp (in space and level) to the CRs as follows:

1. Compute d_1 and d_2 from input images I_1 and I_2
2. Compute optical flow (u, v) using d_1 and d_2 as described in Sec. 4.2
3. Compute I_2^w by warping I_2 according to the flow (u, v)
4. Compute the change in illumination b at each pixel as: $b = I_1 - I_2^w$
5. Filter this field b with a median filter
6. Warp d_2 according to the field of 3D vectors: (u, v, b) .



(a) (b) (c) (d)

Fig. 6: **Image pyramid versus CR-pyramid.** The foreground object in (a) is lost at the third level of the image pyramid (b, top), while still distinguishable at the third level of the CR-pyramid (b, bottom). This results in more accurate flow estimation for longer displacements (c). The histogram in (d) shows that in our set of images, the 20-pixel foreground object can be recovered.

We also experimented with including the variable b in the main energy function and optimizing it as we optimize u and v. While this should be the optimal choice in principle, in practice we find that it is computationally considerably more expensive while presenting similar results to our approximation.

5 Experiments

We have built Channel-Flow by replacing the brightness term in Classic+NL with the intent of isolating the contribution of this new formulation. Consequently, here we provide a detailed comparison with Classic+NL. We focus our analysis on the *MPI-Sintel* dataset (both training and testing) because it contains many fast motions of small or thin objects. Below we use mean endpoint error (MEPE) as a measure of accuracy [1].

5.1 Synthetic Experiment

First we evaluate experimentally the core contribution of our technique, which is recovering large motions of small objects. Since it is difficult to create a real dataset of small objects moving fast with ground truth optical flow, we chose to create a synthetic dataset with the properties we want to test. To that end, we take natural images and create a sequence with a small (20 pixels) circular-shaped region in the foreground (Fig. 6(a)). Then for a range of foreground displacements (0-20 pixels), we compute the optical flow and we measure the error. We repeat this for several images.

We compute the optical flow with 4 levels of the pyramid, using Classic+NL and also using our method. Figure 6(b) shows the third level of the Gaussian pyramids for two consecutive frames (top). Note that the foreground object is hard to be distinguished from the background. Even though the texture of the foreground and background were originally different, the blurring of the pyramid has eliminated much of this difference. In the bottom half of Fig. 6(b) we show one of the channels of the CR-pyramid where the foreground was represented, also at the third level. Note that the object is visible as a bright spot near the center of the image in both frames. Clearly this is a trackable feature in the CR pyramid. This illustrates our central hypothesis, that the channel representation pyramid keeps information about different objects separate, allowing us to smooth within a layer and estimate long-range motions.

Of course, this effect depends on the actual image values. We generated 44 such examples with different natural images, each with the same range of displacements. Figure 6(c) plots the MEPE as a function of displacement of the foreground (the background is stationary). As expected, the error of Channel-Flow is below that of Classic+NL, and this is particularly pronounced for motions above 10 pixels; that is, more than half the diameter of the foreground region.

When the methods fail, the errors are large, and these obscure what is going on. Consequently in Fig. 6(d) we report the percentage of times that the flow is accurately estimated (with an MEPE < 1 pixel). When the object moves a distance half its size (10 pixels), we recover the true flow twice as many times as does Classic+NL.

5.2 Constant Albedo Sequences

Experiment description: In this experiment we take the traditional Classic+NL algorithm and simply substitute the traditional brightness constancy by our descriptor constancy term. The **hypothesis** is that using a CR pyramid preserves more information at high levels of the pyramid, thus making the recovery of large motions and motion boundaries more accurate. In other words, we wish to test whether the basin of attraction around the true flow is wider using the CR data term than using the brightness constancy term. **Dataset:** We want to isolate this question, that concerns the optimization landscape, from that of which data term is more accurate or robust to certain phenomena. For this reason we use the albedo sequence of the *MPI-Sintel* training dataset, where pixel values do not change from one frame to the next[1]. Here we did not use the full training set but, rather, sampled a subset at random.

Quantitative results: Numerical results are shown in Table 1. In addition to the traditional mean end point error (MEPE), we report other statistics for further analysis, following the *MPI-Sintel* standard. Categories *matched* and *unmatched* group pixels according to whether they exist in both frames or not, the s- and d- categories group pixels based on their speed and distance to a motion boundary respectively. Further details can be found in the original dataset publication [7]. The table shows that that Channel-Flow produces overall better results than Classic+NL. The two columns where Channel-Flow outperforms Classic+NL by wider margins are: closer to the boundaries ($d10$) and in the large motions ($s40$). These are areas near motion boundaries and fast moving regions. This provides some confirmation, on complex sequences, of our original hypothesis that the CR should be better for these cases.

Qualitative results: Figure 7 shows some examples of the performance of the two methods. Channel-Flow is able to recover certain details or fast motions where Classic+NL fails. On the other hand, Classic+NL produces very nice and smooth flow fields, while Channel-Flow has a stronger tendency towards piece-wise constant fields, presumably due to the data term being very robust to outliers (see Fig. 4).

[1] Here we must use the training set because the test set does not include constant albedo sequences.

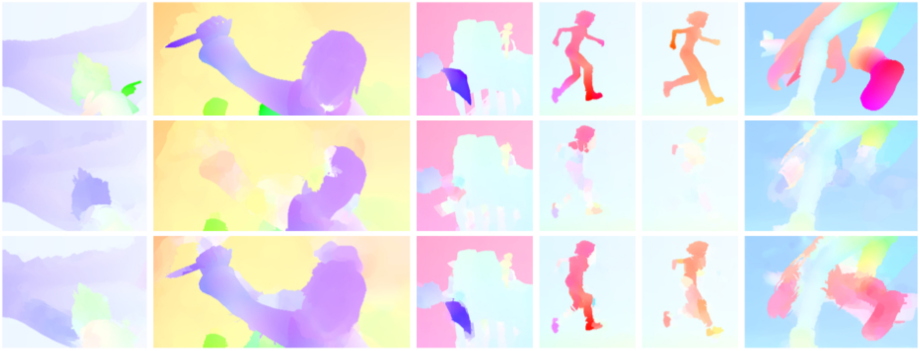


Fig. 7: **Details of results on the albedo training sequences**. **Top row:** Ground truth. **Middle row:** Flow estimation with the traditional approach often fails to capture large motions, especially of smaller objects. **Bottom row:** Using CR's to represent the image improves the accuracy of the flow in such difficult regions. Some examples are (from left to right): Sintel's hair, the arm and knife, the bat's wing, Sintel's limbs, Sintel's body, Sintel's foot.

Table 1: Results on 398 non-consecutive, randomly chosen, image pairs of the albedo sequence of *MPI-Sintel*. Bold letters show the best results in the category.

Method	**MEPE all**	MEPE matched	MEPE unmatched	d0-10	d10-60	d60-140	s10	s10-40	s40
Classic+NL	4.5297	2.6450	**28.857**	5.7011	3.2381	2.0382	0.7703	3.1173	18.7098
Channel-Flow	**4.2178**	**2.1472**	30.923	**4.7790**	**2.5294**	**1.6181**	**0.6936**	**2.2110**	**15.4031**

5.3 Experiments on "Final" Pass of the Sintel Training Set

Experiment description: In this section we test our method in the same set of frames used in the previous section, but this time the frames contain complex phenomena such as changes in illumination, motion blur, fog, etc. The purpose of this experiment is to test the new data term under these additional phenomena. Then, for each of the problems that we identify we propose a solution. The main disadvantage of the CR formulation is that the penalty function is similar for pixel differences that differ by a little or a lot. This problem can be seen in Fig. 4, where the penalty is similar for a change in intensity of 50 and 100. This lack of gradient makes optimization unlikely to converge to the correct solution when there are brightness changes in the scene. To address this we use the illumination model described in Sec. 4.3.

We call the method with the illumination model C-Flow+I in Table 2. In addition, in order to have valid derivatives, the finest level of the CR pyramid needs to be spatially smoothed. Therefore, the two frames are never compared without spatial blur. To refine the output, we use a 1-level pyramid of Classic+NL, and we call this method C-Flow+I+C in Table 2.

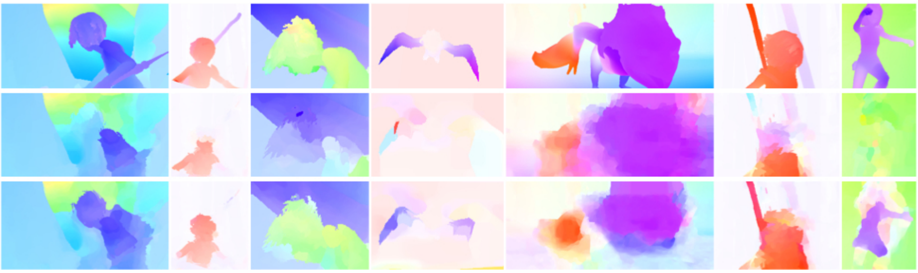


Fig. 8: **Results on the "final" training sequences**. **Top row:** Ground truth. **Middle row:** Results with Classic+NL. **Bottom row:** Results with Channel-Flow and the two additions (C-Flow+I+C)

Results: Qualitative results are shown in Table 2. We observe that both the illumination model and the 1-level pyramid steps improve results in all categories. Visualizations of the recovered change in illumination at each pixel are shown in Fig. 9. We observe that our technique often recovers successfully the low frequency changes in illumination produced by fog, specular reflections, etc. Higher frequency changes are not recovered due to the wide median filter used. Qualitatively, in Fig. 8 we see roughly the same behavior as in the albedo case.

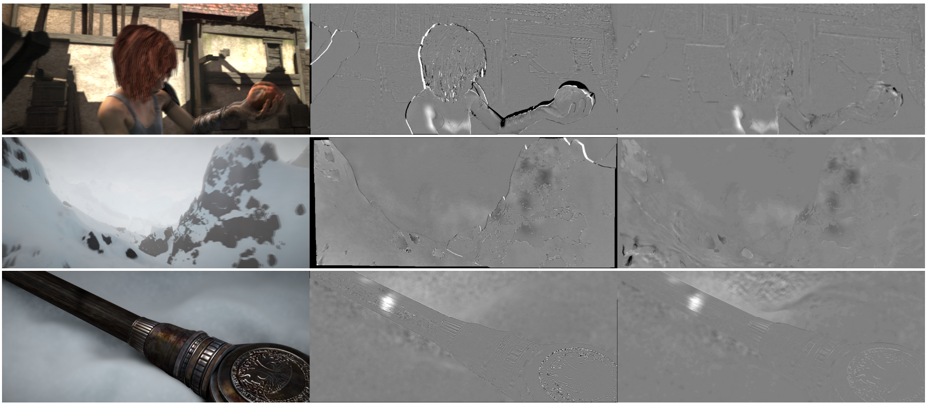


Fig. 9: **Visualization of recovered change in illumination**. **Left:** Average of the pair of input frames. **Middle:** Ground truth change in illumination. **Right:** Estimated illumination change. The ground truth change in illumination is estimated by warping the second frame according to the ground truth flow field, and subtracting this from the first frame. If brightness constancy held, the result would be a constant image of zeros. However, changes in illumination and other complex phenomena (motion blur, smoke, fog, etc) violate this.

Table 2: Results on the same 398 non-consecutive, randomly chosen, image pairs of the final sequence of *MPI-Sintel* . C-Flow+I is Channel-Flow with illumination model. C-Flow+I+C is the same, followed by a refinement with a 1-level pyramid Classic+NL.

Method	**MEPE all**	MEPE matched	MEPE unmatched	d0-10	d10-60	d60-140	s10	s10-40	s40
Channel-Flow	8.0147	6.1496	34.418	9.4287	7.0531	5.4954	1.5578	7.3484	42.2794
C-Flow+I	7.6456	5.8759	30.823	9.3090	6.7750	5.2896	1.4564	6.8913	40.9962
C-Flow+I+C	**7.3330**	**5.5448**	**30.736**	**8.9132**	**6.3343**	**4.9935**	**1.3041**	**6.1424**	**40.1855**

The runtime of the Matlab code for a 1024 × 436 *MPI-Sintel* image pair is about 5 hours on a standard Linux desktop. Half of the computational time is spent on solving the linear equation system using the Matlab built-in backslash function. The parameter values used are: σ_{sp} = 1 and σ_f = 1.2, λ = 100, number of bins = 32, α = 0.45, ϵ = 0.001.

5.4 Experiments on the Sintel Test Set

We evaluate our method on the test set of *MPI-Sintel* and the results are shown in Tables 3 and 4; numbers in parentheses indicate the ranking on the Sintel site at the time of submission. We show only a few methods here; see the MPI-Sintel website for the full comparison and images of our results. Consistent with the experiments above, we see a consistent improvement over the baseline Classic+NL, both in the clean and final sets. In addition we compare with LDOF [4], a popular method for dealing with large displacements. We see that a classical formulation with the CR data term largely outperforms LDOF for large displacements without the use of an external matching process. Since our method is based on Classic+NL, it does not benefit from the latest ideas in optical flow. Other methods like DeepFlow [31] are significantly more accurate, even for large motions. Our results suggest, however, that switching from classical brightness constancy to some sort of descriptor constancy may be valuable and we hypothese that this idea will apply to other methods as well.

Table 3: **Select results on the *MPI-Sintel* test set** for the clean pass. The simple change in data term improves results over the Classic+NL baseline. See the Sintel website for the full table.

Method	**MEPE all**	MEPE matched	MEPE unmatched	d0-10	d10-60	d60-140	s10	s10-40	s40
DeepFlow (6)	5.377	1.771	34.751	4.519	1.534	0.837	0.960	2.730	33.701
Channel-Flow (13)	7.023	3.086	39.084	5.411	3.236	1.918	0.624	2.791	49.021
LDOF (15)	7.563	3.432	41.170	5.353	3.284	2.454	0.936	2.908	51.696
Classic+NL (16)	7.961	3.770	42.079	6.191	3.911	2.509	0.573	2.694	57.374

Table 4: **Select results on the *MPI-Sintel*** test set for the final pass. Channel-Flow again improves over the baseline. See the Sintel website for the full table.

Method	**MEPE all**	MEPE matched	MEPE unmatched	d0-10	d10-60	d60-140	s10	s10-40	s40
DeepFlow (5)	7.212	3.336	38.781	5.650	3.144	2.208	1.284	4.107	44.118
Channel-Flow (13)	8.835	4.754	42.064	6.757	4.566	3.657	1.292	5.349	54.648
LDOF (15)	9.116	5.037	42.344	6.849	4.928	4.003	1.485	4.839	57.296
Classic+NL (16)	9.153	4.814	44.509	7.215	4.822	3.427	1.113	4.496	60.291

5.5 Experiments on the Middlebury dataset

In order to provide a more complete comparison with existing methods, we also test our method in the training set of the *Middlebury* dataset. As we do in the previous experiments, we use the parameter configuration reported by Classic+NL. The MEPE of CR-Flow is 0.287 and the MEPE of Classic+NL is 0.257. We test the statistical significance of these two results and we find them not to be significant. Note that Middlebury motions are small and the value of the CR term for dealing with large motions is not evident here.

6 Conclusion

One of the dilemmas of optical flow is that there is a trade-off between the size of the objects and the magnitude of motions that can be estimated. The large motions and complexity of the *MPI-Sintel* optical flow database demand that such trade-offs be addressed. We have shown how to at least partially address this issue by introducing a channel representation to replace the images used in standard methods. This representation maintains more of the image information under significant blurs.

Our paradigm works with the Classic+NL framework and changes only the data term. This allows us to isolate the effects of this term from other properties of a flow method. We have demonstrated quantitative improvement over the baseline method for a controlled experiment of small regions moving quickly. We have also demonstrated improvement over baseline for the *MPI-Sintel* albedo sequences where brightness constancy holds (except at occlusion boundaries). Given that many of the top-performing methods are based on the variational approach, the channel representation may be potentially very useful for many other flow algorithms as well.

Finally we introduced a simple method to deal with changing brightness, which extends the Channel-Flow method to more complex sequences. This simple method could also be used for other flow algorithms. On the difficult *MPI-Sintel* final test set we show improvement over the baseline, especially in areas near motion boundaries and in fast moving regions.

Acknowledgement DS has been partially supported by NSF grant OIA 1125087.

References

1. 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) (Mar 2011), http://dx.doi.org/10.1007/s11263-010-0390-2 9
2. Berg, A.C., Malik, J.: Geometric blur for template matching. In: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. vol. 1, pp. I–607. IEEE (2001) 4
3. Black, M.J., Anandan, P.: The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. Computer Vision and Image Understanding 63(1), 75–104 (Jan 1996) 1, 4
4. Brox, T., Malik, J.: Large displacement optical flow: Descriptor matching in variational motion estimation. PAMI 33(3) (2011), http://lmb.informatik.uni-freiburg.de//Publications/2011/Bro11a 3, 4, 13
5. Bruhn, A., Weickert, J., Schnörr, C.: Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. IJCV 61(3), 211–231 (2005) 1, 7
6. Burt, P.J., Adelson, E.H.: The Laplacian pyramid as a compact image code. IEEE Transactions on Communications 31(4), 532–540 (1983) 1, 3
7. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: A. Fitzgibbon et al. (Eds.) (ed.) European Conf. on Computer Vision (ECCV). pp. 611–625. Part IV, LNCS 7577, Springer-Verlag (Oct 2012) 1, 2, 3, 10
8. Charbonnier, P., Blanc-Feraud, L., Aubert, G., Barlaud, M.: Two deterministic half-quadratic regularization algorithms for computed imaging. In: IEEE Int. Conf. Image Proc. (ICIP). vol. 2, pp. 168–172 (1994) 6
9. Felsberg, M.: Spatio-featural scale-space. In: Scale Space and Variational Methods in Computer Vision (SSVM), pp. 808–819. LNCS 5567, Springer-Verlag (2009) 4
10. Felsberg, M.: Adaptive filtering using channel representations. In: Mathematical Methods for Signal and Image Analysis and Representation, pp. 31–48. Springer-Verlag (2012) 4
11. Felsberg, M., Forssén, P.E., Scharr, H.: Channel smoothing: Efficient robust smoothing of low-level signal features. PAMI 28(2), 209–222 (2006) 4
12. Granlund, G.H.: An associative perception-action structure using a localized space variant information representation. In: Proceedings of the Second International Workshop on Algebraic Frames for the Perception-Action Cycle. pp. 48–68. AFPAC '00, Springer-Verlag, London, UK, UK (2000), http://dl.acm.org/citation.cfm?id=646050.677227 4
13. Haussecker, H.W., Fleet, D.J.: Computing optical flow with physical models of brightness variation. IEEE Trans. Pattern Anal. Mach. Intell. 23(6), 661–673 (Jun 2001), http://dx.doi.org/10.1109/34.927465 8
14. Horn, B.K., Schunck, B.G.: Determining optical flow. Tech. rep., Massachusetts Institute of Technology, Cambridge, MA, USA (1980) 3
15. Jonsson, E., Felsberg, M.: Accurate interpolation in appearance-based pose estimation. In: Image Analysis, Lecture Notes in Computer Science, vol. 4522, pp. 1–10. Springer-Verlag (2007), http://dx.doi.org/10.1007/978-3-540-73040-8_1 4
16. Jonsson, E., Felsberg, M.: Efficient computation of channel-coded feature maps through piecewise polynomials. Image and Vision Computing 27(11) (2009) 4
17. Koenderink, J.J., Van Doorn, A.J.: The structure of locally orderless images. International Journal of Computer Vision 31(2-3), 159–168 (1999) 4
18. Liu, C., Yuen, J., Torralba, A., Sivic, J., Freeman, W.T.: Sift flow: Dense correspondence across different scenes. In: ECCV (2008), http://dx.doi.org/10.1007/978-3-540-88690-7_3 3

19. Mears, B., Sevilla-Lara, L., Learned-Miller, E.: Distribution fields with adaptive kernels for large displacement image alignment. In: BMVC. IEEE (2013) 4
20. Nordberg, K., Granlund, G., Knutsson, H.: Representation and Learning of Invariance. Report LiTH-ISY-I-1552, Computer Vision Laboratory, SE-581 83 Linköping, Sweden (1994) 4
21. Oron, S., Bar-Hillel, A., Levi, D., Avidan, S.: Locally orderless tracking. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. pp. 1940–1947. IEEE (2012) 4
22. Sevilla-Lara, L., Learned-Miller, E.: Distribution fields. Tech. rep., UMass Amherst (2011)
23. Sevilla-Lara, L., Learned-Miller, E.: Distribution fields for tracking. CVPR (2012) 4
24. Snippe, H.P., Koenderink, J.J.: Discrimination thresholds for channel-coded systems. Biological Cybernetics 66(6), 543–551 (1992) 4
25. Steinbrucker, F., Pock, T., Cremers, D.: Large displacement optical flow computation withoutwarping. In: ICCV (2009) 3
26. Steinbruecker, F., Pock, T., Cremers, D.: Advanced data terms for variational optic flow estimation. In: Proceedings Vision, Modeling and Visualization (2009) 3
27. Sun, D., Roth, S., Black, M.J.: A quantitative analysis of current practices in optical flow estimation and the principles behind them. International Journal of Computer Vision (IJCV) 106(2), 115–137 (2014) 2, 6, 7
28. Sun, D., Roth, S., Lewis, J., Black, M.J.: Learning optical flow. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) European Conf. on Computer Vision, ECCV. LNCS, vol. 5304, p. 83–97. Springer-Verlag (Oct 2008) 1, 7
29. Van Ginneken, B., ter Haar Romeny, B.M.: Applications of locally orderless images. In: Scale-Space Theories in Computer Vision, pp. 10–21. Springer (1999) 4
30. Weber, J., Malik, J., Devadas, S., Michel, P.: Robust computation of optical flow in a multiscale differential framework. IJCV 14 (1994) 3
31. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: Deepflow: Large displacement optical flow with deep matching. In: ICCV. pp. 1385–1392 (2013) 3, 13
32. Werlberger, M.: Convex Approaches for High Performance Video Processing. Ph.D. thesis, Institute for Computer Graphics and Vision, Graz University of Technology, Graz, Austria (June 2012), http://gpu4vision.icg.tugraz.at/papers/2012/werlberger_phd.pdf 8