Segmenting Video Into Classes of Algorithm-Suitability

Oisin Mac Aodha[1,2] Gabriel J. Brostow[1] Marc Pollefeys[2]
[1]University College London [2]ETH Zurich
http://visual.cs.ucl.ac.uk/pubs/algorithmSuitability

Abstract

Given a set of algorithms, which one(s) should you apply to, i) compute optical flow, or ii) perform feature matching? Would looking at the sequence in question help you decide? It is unclear if even a person with intimate knowledge of all the different algorithms and access to the sequence itself could predict which one to apply. Our hypothesis is that the most suitable algorithm can be chosen for each video automatically, through supervised training of a classifier. The classifier treats the different algorithms as black-box alternative "classes," and predicts when each is best because of their respective performances on training examples where ground truth flow was available.

Our experiments show that a simple Random Forest classifier is predictive of algorithm-suitability. The automatic feature selection makes use of both our spatial and temporal video features. We find that algorithm-suitability can be determined per-pixel, capitalizing on the heterogeneity of appearance and motion within a video. We demonstrate our learned region segmentation approach quantitatively using four available flow algorithms, on both known and novel image sequences with ground truth flow. We achieve performance that often even surpasses that of the one best algorithm at our disposal.

1. Introduction

Standard data sets help us measure overall progress for specific computer vision challenges such as object recognition, stereo, feature description, and optical flow. Data sets with good variety help highlight both generalist "winners," and special-purpose algorithms with winning strategies for specific situations. These evaluations are useful for researchers planning a new strategy, but practitioners can have trouble capitalizing on these rankings. A practitioner looks to these scores when picking which algorithm to trust for processing a new set of images. We propose a meta-algorithm, that saves the practitioner from this forced decision, and balances the advantages of generalist vs. special purpose algorithms.

To a limited extent, each algorithm could be used to self-assess its own performance, as is typically done for stereo and optical flow. Most algorithms seeking to optimize a non-convex energy term at test time, know only that once converged, a local minimum has been reached. The room for doubt increases if multiple algorithms, whether competing or collaborating, are solving the same problem using different energy functions or priors. Each "expert" will be satisfied, reporting with its own confidence that it has reached an optimum. Our proposed approach addresses situations in general, where some form of gold standard is available in a training stage, but not at test time.

One example of such situations, and the one we use throughout this paper as a test case for our meta-algorithm, is the optical flow problem. To solve this task, we consider a set of k constituent algorithms working in parallel, and treat them as black boxes. Although it is difficult or expensive to obtain ground truth flow data, just enough is available ([2], [14]) to allow some supervised training. We do not train a linear gold standard measure of success that would calibrate the different algorithms' confidences against each other globally, or pairwise like [24]. Instead, we seek out the correlation between good performance by a constituent algorithm, and specific local situations that can be discerned statistically from the image sequence.

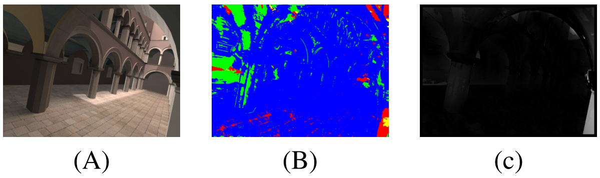

Figure 1. **Per-pixel segmentation results.** A) The first image from a two frame sequence (#18 Sponza1) which is processed as test input to our pre-trained classifier. B) The classifier segments the image, predicting where each of the four algorithms should give the most accurate estimate of optical flow. Color-coding of the pixels indicates the predicted class i.e. algorithm: red=BA [4], green=TV [26], yellow=HS [12], and blue=FL [27]. C) The end-point error for the predicted flow. While the average end-point error based on our prediction does not reach the optimal possible score (0.988 vs. 0.762 pixels), it outperforms the single best algorithm (TV) at our disposal (1.006 pixels).

The semantic segmentation community has been developing successful techniques to find correlations between object-classes and appearance (*e.g.* [10] and [8]). We pose our algorithm-suitability problem as a form of "semantic" segmentation, where the best meaning or label to associate with a given pixel is the algorithm that is predicted to yield the best accuracy. We assume that implementations of all the algorithms under consideration are available, and that instead of speed, accuracy is the only relevant objective. Recognizing that not all flow algorithms have been ported to leverage GPU processing, we accept the fixed cost of running all of them on a given sequence as acceptable, in pursuing the highest accuracy. For choosing among optical flow algorithms (or in a later experiment, descriptor-matching strategies), we speculated that this choice could reasonably be correlated with the contents of *both* images in a sequence. For this reason, our feature vector is computed by analyzing both spatial and temporal characteristics.

This research makes the following main contributions:

- a general segmentation framework for learning to separate pixels by algorithm-suitability,
- improved accuracy for optical flow computed using known constituent algorithms (see Figure 1), and
- a system for easily producing synthetic ground truth data for both optical flow and descriptor-matching.

Experiments show that a special case of the new segmentation approach achieves our initial goal of deciding which algorithm is most suited for which part of a video. We also demonstrate how our segmentation can be used to classify which pixels should *not* be expected to yield accurate optical flow estimates (or descriptor matches).

2. Related Work

Raykar *et al.* [19] describe a generic iterative strategy for computing a standard by which to measure multiple experts, when no gold standard exists. The technique is an improvement over following the majority vote when some experts are better than others. Our problem formulation is different, however, because we cannot assume that one expert is consistently better or worse, independent of the image data being considered.

Learned algorithm selection is shown by Yong *et al.* [29] for the specific task of image segmentation. They used an SVM for learning and performed their experiments on 1000 synthetic images of 2D squares, circles, *etc.,* with additive noise, demonstrating what is actually online learning for algorithm selection. Working with 14 constituent real-time tracking algorithms, Stenger *et al.* [24] developed a learning framework that trained the expected error of each algorithm, given its confidence values. Then during testing, the best-performing pairs of algorithms could be cascaded or run in parallel to track a hand or head. This approach is very flexible for situations where one task is being accomplished at a time. Alt *et al.* [1] describe a supervised learning approach for assessing which planar patches will be difficult for tracking. Using this pre-selection of reliable templates they report an improved detection rate for an existing tracking-by-detection system.

Our situation is different in that various image regions need attention from different algorithms simultaneously. The region-segmentation and labeling of Shotton *et al.* [23] is a closer fit to our needs for dense and parallel k-way classification. Where they have color features as the input data, we use a bank of spatiotemporal features, and their learning algorithm was a forest of extremely randomized trees [9], while we use the simple Random Forests of [5].

Muja and Lowe [18] have presented a unique approach to algorithm-selection, that is quite valuable in the context of feature matching and beyond. Similar to us, they argue that algorithm-suitability is data-dependent. Their system searches a parameter space, where the algorithm itself is just one of the parameters, to find an approximate nearest-neighbor strategy (algorithm and settings). The automatically determined strategy is based on the target data itself, such as a database of SIFT descriptors [15], and desired preferences for optimizing lookup speeds versus memory. There, the training data is the same as the test data, so their optimization is deterministic, while our algorithm suitability must be learned, so we can predict which segments are suited for which strategy, just by looking at a video.

Optical Flow Of the existing approaches to computing optical flow, the iterative FusionFlow [13] is still very different technically, but the closest to our approach in terms of its philosophy. They compute a discrete optimization on continuous-valued flow-fields (with another continuous optimization "clean-up"), by performing a minimal cut on an extended graph. The extended graph consists of auxiliary binary-valued labels to represent either accepting a newly proposed flow vector at that location, or keeping the current flow estimate. The similarity to our work is that in each such iteration of FusionFlow, the new proposed solution could be viewed as a competing strategy or algorithm, offering a potentially lower energy than the current estimate, at least in some spatial neighborhood. Both algorithms benefit from good proposed solutions, but the benefits for our algorithm are more direct (see Section 3.1) and we have no flow-specific pre- or post-processing. FusionFlow is quite flexible and could potentially be modernized with more competitive starting-proposals than the 200+ based on Lucas-Kanade [16] and Horn and Schunk [12], but the authors indicate that because of their energy function, the computed minimum eventually gives a score extremely close to the energy of the ground truth solution (*e.g.* $E = 1613$ vs. 1610).

One advantage of our approach is that no one energy function need be employed universally throughout an image sequence.

A thorough understanding of existing energy functions allowed Bruhn *et al.* [6] to formulate a new Combined Local-Global (CLG) method, aptly named "Lucas/Kanade Meets Horn/Schunk". Their new 2D energy term (and its 3D variant) combined the local robustness to noise offered by algorithms such as Lucas-Kanade [16], with the regularized smoothness and dense flow of global algorithms, such as Horn and Schunk [12]. This approach is still one of the top performers for the Yosemite sequence. Also, they compute a confidence criterion based on this new energy term, and demonstrate that it is partly correlated with actual accuracy. The challenge they describe has been one of our driving motivations, namely, that one has few if any reliable confidence measures, beyond the chosen energy function itself. That problem is compounded when comparing multiple algorithms with different energy-minimization objectives.

The nonparametric FRAME model of Zhu *et al.* [30] optimized its texture synthesis by picking out filters from a filter bank, whose responses are correlated with neighborhoods in the training image. That approach is very flexible, adaptively using potentially many filters, including non-linear ones which filter large sub-images. Since then, Roth and Black's Fields of Experts (FoE) [21] has gained a following by augmenting FRAME, extending Markov random fields with the capability to *learn* filters that model local field potentials. The completely data-driven nature of FoE is very attractive, and Woodford *et al.* [28] showed a method that trains with 5x5 cliques in a comparatively short time. Roth and Black have further demonstrated FoE for the purpose of modeling an optical flow prior [20]. In [20], they used range-images of known scenes with separately obtained real camera motions to learn a model of motion fields, which are different from optical flow. Here, they still had to manually monitor convergence of the learning, but in testing, demonstrated superior results using these spatial statistics as priors for the aforementioned 2D Bruhn *et al.* [6] flow algorithm. FoE's expert functions are less flexible than the FRAME model by design: they can be non-linear, but need to be continuous, and the log of an expert has to be differentiable with respect to both the expert's parameters and the (necessarily) linear filter responses. For our task however, non-linear "filters" are used for computing posteriors (see Section 4).

Sun *et al.* [25] adapted their spatial FoE model of optical flow, learning a relationship between image and flow boundaries, this time with a parameterization of spatiotemporal brightness inconstancy. The steered model of flow and the generalized data term are learned on the painstakingly prepared ground truth flow data of Baker *et al.* [2]. In our experiments, we too train on this data and also have no need for sequence-specific parameter tuning, and we achieve better scores simply by virtue of leveraging multiple black-box algorithms that are effective in their own right.

An important result of the FoE line of research is the finding, that with careful optimization procedures, a good generalist algorithm's priors about local responses to linear filters should be learned from representative training data. Different low-dimensional "experts" in this setting are not unique algorithms, but are instead measures, being combined to model high dimensional probability distributions of parameterized statistics. Our goal is much simpler, non-parametric, and complementary: to establish the *discriminability* between visual situations given competing strategies or algorithms, in this case, for computing optical flow. For example, the algorithms with FoE-based priors trained with different sized cliques (5x5 for [20], 9x9 for [25]) could be evaluated as different strategies in our framework.

The online Middlebury Optical Flow Evaluation [2] currently ranks over 30 algorithms. To test our hypothesis, we need a set of these as constituent algorithms, where they each have at least some unique properties. For simplicity, we chose the four algorithms for which an author's implementation was most readily available: Werlberger *et al.*'s FlowLib [27], Wedel *et al.*'s TV-L1-improved [26], Black and Anandan [4], and Horn and Schunck [12]. For brevity, we will refer to them as FL, TV, BA, and HS, respectively.

3. Algorithm-Selection as Segmentation

The applicability of most algorithms is situation-specific, and we wish to classify those situations automatically. Interest point detectors, such as Good Features to Track [22] or Harris-Stephens corners [11], are already a form of binary segmentation for algorithm-selection purposes. Local neighborhoods are labeled as either regions of interest or not, depending on their appearance in *one* frame. Such segmentations have the expectation that only certain regions can be reliably tracked or matched in subsequent frames.

We expand on this binary algorithm-suitability philosophy with a probabilistic formulation that allows for a k-way pairing: accuracy should be improved if each part of an image sequence is handled by the most suitable of k algorithms. The proposed approach is most appropriate in situations where either no good single algorithm exists, or where a generalist algorithm makes mistakes in places that some specialist algorithm does not. Ideally, we could discern these situations correctly, or could assign a $(k+1)$'th label in places where no algorithm is suitable, producing a complete segmentation.

To reiterate, the single classifier is taking the place of the multiple algorithm-specific energy terms or confidence measures. Being probabilistic, the posteriors of different classifiers can be compared to each other, which is shown to be especially useful when $k = 2$.

3.1. Choice of Learning Algorithm

The classification algorithm for this approach needs to be multiclass, able to handle large amounts of data, and be robust to noise. We have selected the Random Forests algorithm developed by Breiman [5]. We also experimented with Boosted Trees and SVMs and noted slightly worse performance and an increase in training time. Random Forests is an ensemble of decision trees which averages the predictions of the trees to assign the class labels. It makes use of bagging to uniformly sample (with replacement) subsets from the dataset to train the decision trees. It then uses the remaining data to estimate the error for that particular tree. During training, each node in the tree is presented with a random subset of the feature vector. Random Forests have the advantage of being fast to train and inherently parallelizable. Additionally, they can handle large datasets and also estimate the importance of the input variables. Caruana *et al.* [7] provide a thorough evaluation of Random Forests on high dimensional data.

4. Available Features

Algorithm selection is posed as a standard supervised learning problem with training data of the form:

$$\mathcal{D} = \{(\mathbf{x}_i, c_i) | \mathbf{x}_i \in \mathbb{R}^d, c_i \in \mathbb{Z}^k\}_{i=1}^n \quad (1)$$

with n being the number of training examples, k the number of algorithms and d the dimensionality of the feature vector. When constructing a feature vector for optical flow, the goal is to include any image features that could be indicative of places where the different algorithms succeed and break down. Given an image pair I_1 and I_2 (where $I = f(x, y)$ is a grayscale image), a feature vector is computed for each pixel in the first image. While certainly not exhaustive, the following combined single and multi-frame information used in our prototype feature set already produces good results:

Appearance: Highly textured regions provide little challenge for modern optical flow algorithms. By taking the gradient magnitude of the image it is possible to measure the level of "texturedness" of a region:

$$g(x, y, z) = ||\nabla I_1||, \quad (2)$$

The notation $g(x, y, z)$ indicates that the function is evaluated at an x, y position and level z in the image pyramid. Additionally, the distance transform is calculated on edge detected images:

$$d(x, y, z) = disTrans(||\nabla I_1|| > \tau), \quad (3)$$

The intuition is that image edges may co-occur with motion boundaries, and the higher the distance from them, the lower the chance of occlusion. We also experimented with other appearance features such as the convolution with banks of Gabor features to capture neighborhood information. Qualitatively, they did not contribute to the overall results and so were excluded.

Time: Some flow algorithms can break down at motion discontinuities. Identifying these regions could be a cue for improving flow accuracy. One method to find these regions is to do simple image differencing. Through experimentation, we found that a more robust approach was to take the derivative of the proposed flow fields. This is done by taking the mean of the different candidate algorithms' flow and calculating the gradient magnitude in the x and y directions, $t_x = ||\nabla \bar{u}||$ and $t_y = ||\nabla \bar{v}||$ respectively.

Photo Constancy: One estimate of confidence for an optical flow algorithm is to measure the photoconsistency residual r. This is done by subtracting the intensity in the second image from the first at the predicted location of the pixel. Due to the discrete nature of image space, we perform bicubic interpolation to interpolate the intensity values in the second image. The residual error, measured in intensity, is calculated independently for each of the k candidate flow algorithms:

$$r_i(x, y, k) = I_1(x, y) - bicubic(I_2(x + u_i(k), y + v_i(k))) \quad (4)$$

Scale: Most effective approaches to optical flow estimation utilize scale space to compute flow for big motions. With this in mind, all of these features, with the exception of the residual error, are calculated on an image pyramid with $z = [1, 10]$ levels and a rescaling factor of 0.8.

The result for optical flow is a 44 dimensional feature vector $\mathbf{x}_i$, computed for each of the pixels in the first image:

$$\begin{aligned} \mathbf{x}_i &= \{g(x, y, [1, 10]), d(x, y, [1, 10]), t_x(x, y, [1, 10]), \\ &\quad t_y(x, y, [1, 10]), r(x, y, [1, k])\} \end{aligned} \quad (5)$$

5. Synthetic Data

Several techniques have been proposed to generate ground truth optical flow data from real image sequences. The popular Middlebury optical flow dataset approximated flow by painting a scene with hidden fluorescent texture and imaging it under UV illumination [2]. The ground truth flow is then computed by tracking small windows in the high resolution UV images, and performing a brute-force search in the next frame. The high resolution flow field is then downsampled to produce the final ground truth. This technique, while successful, is extremely time consuming and restrictive in the types of scenes that can be captured (restricted to lab environments). Liu *et al.* [14] use human assistance to annotate motion boundaries in image sequences as a preprocessing step for layer-wise optical flow estimation. While this approach is more practical, it still relies on the accuracy of the flow estimation algorithm used.

Synthetically generated sequences have been used as an alternative to natural images for optical flow evaluation since the introduction of the famous Yosemite sequence by Barron *et al.* [3]. These have the advantage of being quick to generate and can benefit from advances in computer graphics to create highly realistic scenes. Using complex geometry, detailed texture, and global illumination techniques, it is now possible to generate realistic sequences with consumer 3D computer graphics packages. Currently there is no publicly available way to generate ground truth data of high resolution photo-real synthetic images.

We have developed a plugin for Maya which allows the user to generate ground truth optical flow for a given image pair. An example output of the system is shown in Figure 2. The system works by casting a ray from the camera center, through the image plane and into the scene until it intersects an object. Then, this point is projected back into the second camera (respecting occlusions in the scene) and the optical flow is calculated from the position difference with respect to the first image plane. An advantage of the system is that the texture and lighting of the scene is independent of the geometry. This creates the possibility for re-rendering the same scene using different illumination and textures, without altering the ground truth. As the system calculates intersections between projected rays and scene objects, occlusions are noted and therefore not erroneously labeled with incorrect flow (black regions in Figure 2). Additionally, there is no restriction on camera baseline or object motion and the same system can create data for image descriptor evaluation. The generated sequences and the software to create additional ground truth optical flow for experiments is available online.

6. Experiments

Optical Flow In all the experiments, we used the same learning system described in Section 3.1, and unless otherwise stated, $k = 4$, so we are predicting which one of the four constituent optical flow algorithms (BA, TV, HS, FL) to trust at each pixel. The algorithms were used with their default or most-successful published settings, though in practice, the same algorithm with different parameters could be treated by our meta-algorithm as distinct classes.

For comparison purposes, we designed a baseline hybrid from our own experiences of using these algorithms. Having analyzed the results of the four algorithms on all eight Middlebury training sequences, we noted in particular the performance for areas of low and high texture, and motion discontinuities. While FL performs the best overall, BA was the best in areas of high texture. Texture is measured simply by thresholding the gradient magnitude of the image, $||\nabla I_1|| > t$, and dilating the result with a 5 pixel ball structuring element to mask which pixels would be associated with "class" BA. With a threshold of $t = 10$, the new algorithm, called the Trivial Combination (TC), was also used in our tests.

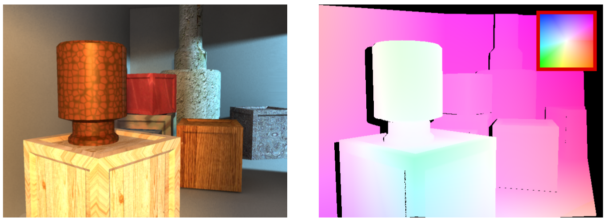

Figure 2. **Unlimited Ground Truth Optical Flow Data** A) Example frame from a two frame sequence (#9 Crates1), generated by our system. The scene was modeled in Maya and rendered using Mental Ray to include global illumination. B) Ground truth optical flow between the two frames. The flow field color is displayed inset and coded using the same format as [2]. Black values in the flow image indicate areas of occlusion between the two frames.

We used image pairs with ground truth flow from three sources: the eight Middlebury training sequences [2], two Middlebury-like sequences from [25], and nine of our own challenging synthetic sequences, for a total of 19. In line with the evaluation of [2], the reported scores are the average end point error across the whole image. The error is not reported for areas known to have no flow, and for a 10 pixel boundary region around the edge of the image. Table 1 lists the main results for our algorithm, and additional illustrations of the performance of the different learning parameter settings are in the Supplemental Materials, as are qualitative results of flow computed on real images without ground truth.

Leave-one-out tests were performed to make the best use of available training data. Due to the redundancy present when multiple algorithms could give the correct flow for a pixel, we pre-select a subset of the available data on which to train. We only train on examples where the end point error between the best performing algorithm and the second best for a particular pixel is greater than a threshold of 0.3 pixels. This reduces the amount of training data but also allows the selection of examples which are most discriminative. It also prevents training on areas of similar error. The trained forest reveals feature usage that is almost equal throughout the feature vector. Balanced usage of features indicates that none of our available features is especially better or worse than the others.

We did a separate experiment to compute our results on the Middlebury *test* sequences. We trained our system on just the eight Middlebury sequences, and our reported end point errors are as follows, Army: 0.1, Mequon: 0.35, Schefflera: 0.63, Wooden: 0.22, Grove: 0.91, Urban: 0.62, Yosemite: 0.15, and Teddy: 0.74. These results place us between "Occlusion bounds"' and "Multicue MRF" on the online Middlebury table, which means we appear to outperform BA and HS, but not TV or FL on that data set. This shortfall compared to two of our constituent algorithms is surprising, though our implementations of FL and TV are known to differ from those used in the competition. Interestingly, we currently outperform the FoE and other hybrid algorithms described in Section 2.

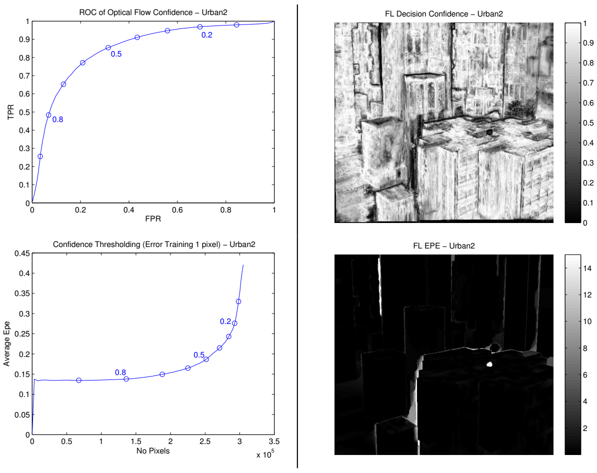

Figure 3. **FL Decision Confidence.** Left) the top figure shows the ROC curve of confidence in FL's optical flow for computing a pixel flow of less than 1 end point error for the Urban2 sequence. The bottom left figure shows that by choosing a high probability, the average end point error is kept low, at the expense of the number of pixels where flow was estimated. The classifier allows the user to trade off error versus coverage. Right) on the top, the same confidences are plotted spatially, revealing high uncertainty near object discontinuities and at the image boundaries. The bottom plots the true FL end point error for all pixels.

Another test of our hypothesis involved trying the $k = 2$ case, where one algorithm is the leading optical flow strategy in our cohort, FL, and the other "strategy" is to not report flow for a given pixel, expressly avoiding making a flow estimate whose error is anticipated to be too big (1 pixel end point error in this case). A perfect classifier would only assign the FL label to pixels that would be estimated to within one pixel of the true vector. This test was also done as leave-one-out from our set of 16 sequences, and the plot of the Urban2 sequence in Figure 3 is a representative of the other tests. The main benefit of this binary classifier is that the confidence is now measured as the probability of computing a pixel flow of less than 1 pixel end point error. Had we used the algorithm's own energy function, the confidence could not be directly compared to a similar binary classifier for each other flow algorithm.

Our unoptimized code for the classifier is implemented in C, though features and other visualization functions are computed in MATLAB. Running on an Intel Core 2 Quad 2.4Ghz with 3.25Gb RAM, computing all the flow estimates takes approximately 5min. for the 640x480 images we tested, and generating the feature vector takes another 1 min. Random Forest takes 1hr 15mins to train on all our data in the leave-one-out experiments. However, once it is trained and the features are computed, it takes on the order of seconds to compute a prediction.

Feature Matching Using the same self evaluation strategy as in optical flow, we conducted an experiment on feature matching. We trained our algorithm on images from nine different scenes, ranging from two to 10 images per scene. The images are made up of sequences from Mikolajczyk's [17] dataset and our own synthetically generated training images. The scenes exhibit changes due to rotation, scale, image blur, affine transformation, illumination and large viewpoint changes. 65,000 interest points were extracted using the Harris-Hessian-Laplacian interest point detector and described using the SIFT algorithm. For each SIFT descriptor, a feature vector was computed, consisting of the Euclidean distance ratios between the first four closest matched features in the second images and in the same image. The intention of using this self similarity measure was to reduce false positives due to repeating structures. For the scenes from Mikolajczyk, ground truth correspondence is provided by a homography relating each image pair. For our synthetic data, we have the ground truth correspondences by virtue of our system. Correspondence is measured against the nearest neighbor distance ratio test of Lowe [15]. Normally, given two features from separate images, they are considered a match if the distance ratio between the first and second closest features is below a threshold and the position of the feature in the second image is less than 1.5 pixels from the ground truth. Mikolajczyk also use a region overlap error to verify that the descriptors are describing the same area in the images, but due to the large viewpoint changes present in our data, this is not applicable. We tested our classifier on 5000 features from the graffiti sequence [17]. Figure 4 shows the ROC curve obtained by sweeping the probability given by the classifier. It also displays the performance of SIFT matching by varying the distance ratio threshold t. The curve shows that the classifier outperforms the standard SIFT matching criterion.

7. Conclusions

The results of the learning experiments indicate that at least for optical flow and feature matching, it is indeed pos-

Image Sequence		BA	TV	HS	FL	TrivComb	Ours	OptCombo
1	**Venus**	0.445	0.408	0.549	0.350	0.387	**0.344**	0.271
2	**Urban3**	0.892	1.132	1.329	**0.527**	0.589	0.603	0.398
3	**Urban2**	0.552	0.506	0.772	**0.435**	0.428	0.436	0.256
4	**RubberWhale**	0.148	0.135	0.182	**0.096**	0.133	0.097	0.074
5	**Hydrangea**	0.217	0.196	0.276	**0.164**	0.191	0.165	0.123
6	**Grove3**	0.705	0.745	0.873	**0.622**	0.670	0.628	0.466
7	**Grove2**	0.197	0.220	0.285	**0.170**	0.189	0.171	0.111
8	**Dimetrodon**	0.161	0.211	0.171	**0.144**	0.147	0.147	0.115
9	**Crates1***	4.439	**3.464**	5.195	3.724	4.582	3.748	2.448
10	**Crates2***	17.517	**4.615**	17.891	12.634	17.634	9.607	3.373
11	**BrickBox1***	7.059	7.025	7.129	**6.748**	7.045	6.791	6.606
12	**BrickBox2***	21.820	21.868	**21.643**	22.333	21.784	21.9	20.925
13	**Mayan1***	2.186	2.331	2.702	**0.709**	2.177	1.833	0.443
14	**Mayan2***	0.449	0.442	0.876	0.344	0.389	**0.342**	0.212
15	**YosemiteSun**	**0.221**	0.310	0.267	0.250	0.221	0.252	0.186
16	**GroveSun**	0.624	0.576	0.683	**0.403**	0.474	0.437	0.368
17	**Robot***	8.55	8.959	**8.443**	8.607	8.52	8.612	8.11
18	**Sponza1***	1.06	1.006	1.281	1.021	1.057	**0.988**	0.762
19	**Sponza2***	**0.436**	0.496	0.464	0.473	0.456	0.467	0.296

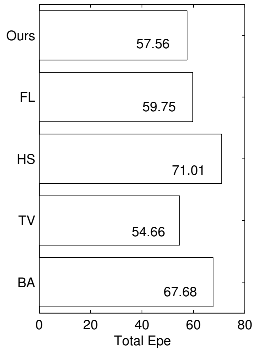

Table 1. **Leave-1-Out Tests** Results of leave-one-out end-point pixel error testing of flow with 19 image pairs. Sequences marked with an asterisk were generated as described in Section 5. Rows show how each optical flow algorithm fared on a given test-sequence: BA=Black and Anandan [4], TV=Wedel *et al.*'s TV-L1-improved [26], HS=Horn and Schunck [12], FL= Werlberger *et al.*'s FlowLib [27], TrivCombo= our baseline hybrid, Ours= our proposed meta-algorithm, and OptCombo= the lowest error attainable with a discrete combination of all four constituent algorithms. The lowest error for each sequence, excluding the optimal, is highlighted in bold.

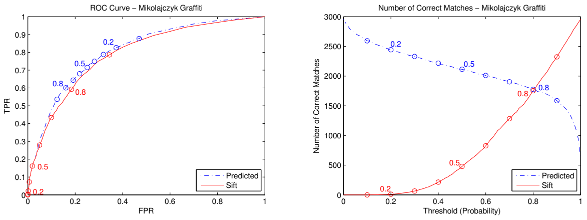

Figure 4. **SIFT Decision Confidence.** Left) ROC curve showing the result of classification for self evaluation for SIFT feature matching. The classification approach performs better than standard SIFT matching strategies that rely simply on a distance ratio, e.g. 0.8. Right) The figure shows the number of correct features matched as the probability from the classifier is increased. It can be seen from the diagram that a conservative threshold on the probability would result in fewer matches.

sible to estimate a correct classification of pixels based on algorithm suitability. Here, optimally correct classification would mean that any segments with the same class label will be best handled by that one-among-k available algorithms. While our predictions about where to use each constituent algorithm only approximate that optimal labeling, we show that on average, results based on our labeling often outperform even a universally-good constituent algorithm working individually.

Algorithms for both the tasks we examined, flow and description, have various criteria for assessing their own performance or confidence. One of the benefits of the proposed framework, at least for practitioners, is that instead of using algorithm-specific heuristics, the goal itself is used with training data, to learn a unified "success classifier." In practice, this means that parts of an image sequence can be identified and excluded automatically and more adaptively, when deemed too hard. This approach applies even to the $k = 1$ case, squeezing more accuracy out of the already good FlowLib and SIFT algorithms.

A secondary contribution of this work is our new system for synthesizing image sequences, producing ground truth suitable for both optical flow and feature description experiments. We are releasing this system for general use to aid research where algorithms could benefit from extended training data. By leveraging existing tools in the computer graphics community, each scene with potentially complex geometry and motion can have one set of ground truth correspondences, while rendering endless variants of the image sequence by changing the lighting and textures. We provide code to run the experiments in this paper, code for generating synthetic data, and sample data on our website. We hope that that this encourages experiments using new and unusual constituent algorithms.

7.1. Limitations & Future Work

While the proposed meta-algorithm can produce better results than the constituent algorithms alone, limitations and new challenges have come to light. The optimal labelings clearly exist, but still elude us. One hope is that further meaningful correlations between the input data and the class labels could be discovered by expanding the variety of features available to the classifier, beyond those proposed in

Section 4.

Building our classifier on a standard Random Forest has proven advantageous in many ways, but limits the utility of some of the training data. As with most discrete multi-class classifiers, each training example specifies that one algorithm is most-suitable, while the rest are equally unsuitable. This effectively ignores the fact that, for optical flow for example, the second-best algorithm could give an end-point estimate 10 times closer than the fourth-best. Equally, when the difference between the top two algorithms is minimal, our current system is forced to either ignore the example completely, or expend effort trying to learn to distinguish between equals.

We chose to validate our hypothesis mostly on the example application of optical flow. There are many other applications, such as stereo, where multiple competing algorithms vie to be universally best, and it would be interesting to try our learned segmentation approach there. Random Forests can handle more than five or six classes in general, but exactly how far our approach will scale is unclear. A particular goal for the future would be the development of new specialist algorithms, which would be terrible on their own, but could nicely complement the generalist algorithms. Our experiments here were designed to be most-faithful to the state-of-the-art, so we simply included the four algorithms for which the authors' implementations were immediately available.

Finally, our approach ignores the cost of processing times, which is currently acceptable, but $O(k)$ in the number of algorithms under consideration. One strategy could be to optimize the forest subject to the computational cost of each algorithm. Overall, the proposed method learns to segment by algorithm-suitability through analyzing features in two frames at a time, which could be used intermittently for long sequences if faster processing is required.

Acknowledgements

We thank the authors who provided reference implementations of their algorithms and data. Funding for this research was provided by the Microsoft Innovation Cluster for Embedded Software.

References

[1] N. Alt, S. Hinterstoisser, and N. Navab. Rapid selection of reliable templates for visual tracking. In *CVPR*, 2010.

[2] S. Baker, S. Roth, D. Scharstein, M. J. Black, J. P. Lewis, and R. Szeliski. A database and evaluation methodology for optical flow. In *ICCV*, pages 1–8, 2007.

[3] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *IJCV*, 12(1):43–77, feb 1994.

[4] M. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow-fields. *CVIU*, 63(1):75–104, January 1996.

[5] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[6] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *IJCV*, 61:211–231, 2005.

[7] R. Caruana, N. Karampatziakis, and A. Yessenalina. An empirical evaluation of supervised learning in high dimensions. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 96–103, New York, NY, USA, 2008. ACM.

[8] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, pages 1778–1785, 2009.

[9] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Mach. Learn.*, 63(1):3–42, 2006.

[10] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009.

[11] C. Harris and M. Stephens. A Combined Corner and Edge Detector. In *4th ALVEY Vision Conference*, pages 147–151, 1988.

[12] B. Horn and B.G.Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[13] V. Lempitsky, S. Roth, and C. Rother. Fusionflow: Discrete-continuous optimization for optical flow estimation. In *CVPR*, pages 1–8, 2008.

[14] C. Liu, W. T. Freeman, E. H. Adelson, and Y. Weiss. Human-assisted motion annotation. In *CVPR*, pages 1–8, 2008.

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

[16] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, pages 674–679, 1981.

[17] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 27(10):1615–1630, October 2005.

[18] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISSAPP (1)*, pages 331–340, 2009.

[19] V. C. Raykar, S. Yu, L. H. Zhao, A. Jerebko, C. Florin, G. H. Valadez, L. Bogoni, and L. Moy. Supervised learning from multiple experts: whom to trust when everyone lies a bit. In *ICML*, pages 889–896. ACM, 2009.

[20] S. Roth and M. J. Black. On the spatial statistics of optical flow. *IJCV*, 74(1):33–50, 2007.

[21] S. Roth and M. J. Black. Fields of experts. *IJCV*, 82(2):205–229, 2009.

[22] J. Shi and C. Tomasi. Good features to track. In *CVPR*, pages 593 – 600, 1994.

[23] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR*, 2008.

[24] B. Stenger, T. Woodley, and R. Cipolla. Learning to track with multiple observers. In *Proc. CVPR*, June 2009.

[25] D. Q. Sun, S. Roth, J. P. Lewis, and M. J. Black. Learning optical flow. In *ECCV*, pages III: 83–97, 2008.

[26] A. Wedel, T. Pock, C. Zach, D. Cremers, and H. Bischof. An improved algorithm for TV-L1 optical flow. In *Proc. of the Dagstuhl Motion Workshop*, LNCS. Springer, September 2008.

[27] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic Huber-L1 optical flow. In *BMVC*, 2009.

[28] O. Woodford, I. D. Reid, P. H. S. Torr, and A. W. Fitzgibbon. Fields of experts for image-based rendering. In *BMVC*, volume 3, pages 1109–1108, 2006.

[29] X. Yong, D. Feng, Z. Rongchun, and M. Petrou. Learning-based algorithm selection for image segmentation. *Pattern Recogn. Lett.*, 26(8):1059–1068, 2005.

[30] S. C. Zhu, Y. N. Wu, and D. Mumford. Filters, random fields and maximum entropy (frame): Towards a unified theory for texture modeling. *IJCV*, 27(2):107–126, 1998.