Edge Descriptors For Robust Wide-Baseline Correspondence

Jason Meltzer Stefano Soatto

Abstract

This paper describes a method for finding wide-baseline correspondences between images at locations along gradient edges. We find edges in scale space using established methods and develop invariant descriptors for these edges based on orientation and scale histograms. Because edges are often found on occluding boundaries, we calculate and store two descriptors per edge, one on each side, for robustness to occlusions. We demonstrate the effectiveness of edge matching in the applications of wide-baseline correspondence, structure from motion from line segments, and object category recognition on the Caltech 101 dataset.

1. Introduction

Image edges have been studied since the early days of computer vision [2] and have been used as low-level features for applications as diverse as structure from motion [19, 3], segmentation [10], and recognition [15]. Despite their widespread utility, no local wide-baseline matching technique is available for finding correspondence between edges in different images, as exists for point features [9, 12, 22, 11]. In this paper, we propose a simple system for matching edges using descriptors, akin to affine invariants for points, and call these *edge descriptors*.

A point feature is anchored to a single position in an image or image scale space, and its descriptor is a function of the image within a region containing that point. SIFT [9], for example, selects a distinctive point in scale space, (x, y, σ) and develops a descriptor based on the position and orientation of gradients taken at the selected scale, σ, and falling within a circle centered at (x, y) with radius proportional to σ. In contrast, edges are one-dimensional contours, commonly thought of as separating regions of constant intensity [2] or constant texture [16]. In the context of the aperture problem, constant-intensity regions are non-descriminative – they are all identical. This may lead one to believe that it is not possible to create local image-based descriptors for edges. However, it is important to note that regions on either side of an edge are only smooth *above a particular scale*, not at other scales. By considering multiple scales, we can indeed develop distinctive local descriptors anchored to edges with similar invariance properties as point-based descriptors.[1]

To further complicate matters, edges are frequently found on occluding boundaries, where a change in viewpoint will violate the hypothesis of affine warping; the image on one side of the occlusion will be inconsistent with changing viewpoint. For point-features, this is fatal – the image inside the support region of the descriptor can change drastically, making matching impossible, and there is no way to determine if the point of interest falls near an occlusion when generating the descriptor (see Figure 3 for an example of this effect). While edges are more likely to fall on occluding boundaries, we know this *a priori*, and there is a simple way to deal with them: separate the domain of the support region into two parts, one on either side of the edge, and develop a descriptor for each of these regions. Using this bipartite descriptor allows us both to be robust to occlusions and to detect them when matching, as only one of the two descriptors will match when an edge falls on an occluding boundary.

Edges naturally provide another key piece of information useful for correspondence: ordering. Though an edge may warp due to changes in viewpoint, the image information along the same edge will never appear "out of order" between viewpoints. It is important to emphasize that the descriptors we present do not explicitly take into account edge geometry[2]; they are primarily based upon image information in a region of scale space split by an edge. Edge descriptors do, however, utilize the edge contour to separate support regions into two sides, and they take into account the ordering of pixels and directionality provided by edge detection and chaining.

A brief summary of our algorithm is outlined here, with details in following sections. Since we are finding descriptors attached to edges, first these edges must be detected in an image (§2.1). We use established techniques to find edges in scale space, which are stored as lists of ordered tuples $(x_t, y_t, \sigma_t, \varsigma_t^+, \varsigma_t^-)$, where a different descriptor scale, ς^+, ς^-, is computed for either side of the edge (§2.3). Along

[1] Of course, this is not true for all edges. Those which separate smooth regions at all scales will not admit a distinctive descriptor along an edge or at any point, for that matter.

[2] Arguments such as those presented in [23] indicate that for any viewpoint invariant statistic, edge geometry is not discriminative.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

any given edge, canonical positions on either side are chosen at the local extrema over these scales (§3.1). At each of these anchors, a histogram of gradient orientations is computed at scales other than σ_t (§2.4). These are stored in an ordered list, which comprises the *edge descriptor.* Since the same edge in differing images will, in general, have a different length, directonality of ordering, and missing parts, we must account for this variability in the matching procedure. We use the Smith-Waterman technique derived from protein sequence matching ([17]) to find the best possible match between any pair of edge descriptors (§3.3).

Experimental results (§4) demonstrate the effectiveness of our edge descriptors in the contexts of correspondence, structure from motion, and category recognition.

1.1. Related work

There is a rich literature covering the concept of edge detection, and we will not presume to review it thoroughly here. Instead call attention to a few citations of direct applicability to this paper. One should first note that there are many ways to define an "edge" in a static image. Following Canny [2], edges separate regions of smooth intensity that ideally differ by a step function in one direction (normal to the edge). An alternative approach is to consider an edge as separating two regions with sufficiently different image statistics. These "texture edges" are commonly computed with the compass operator [16], which finds an edge orientation cutting a circular region such that the two halves of that region have maximally different statistics. Both of these approaches can be made multi-scale, where the scale parameter determines the step size of gradient computations and the support region surrounding a candidate location.

The method of edge detection we have chosen to implement derives from Lindeberg [7]. This is a multi-scale detector based on the zero-crossings of the second directional derivative of the image at a particular scale. Edges are chained by evaluating every voxel in scale-space, assigning it an orientation, and linking adjacent voxels if their positions and orientations are sufficiently close. Each position along an edge is assigned a location, scale, orientation, and edge strength. We use the output of this detector to localize features and develop invariant descriptors.

Recently, Tsin *et al*. [21] have demonstrated a technique for edge *tracking* in video using the randomized forest learning framework [6]. Edges are initially selected by hand, then tracked throughout a video sequence by learning a forest of randomized decision trees, which act as a type of image descriptor around a selected edge. This method requires that the small-baseline assumption hold, as the descriptor incorporates little invariance to wide-baseline deformations.

Mikolajczyk *et al*. [13] find descriptors that are anchored to points along edges found using a multi-scale Canny detector. At a given edge pixel, they select the scale at which a Laplacian filter convolved with the image attains a maximal response. This defines the support region for the point descriptor, which consists of gradient orientations weighted by magnitudes on edge pixels, binned into a corse and fine position grid. The domain of the support region is split in half by finding a dominant orientation, and one descriptor is computed for each side. While [13] and our technique share a number of similarities, one primary difference (beyond details of the descriptor and edge detection) is that [13] considers descriptors as independent; they find point-features that fall on edges, whereas we seek a unified descriptor for an entire edge.

2. Edge Descriptors

As our goal is to match edges in differing images, we must develop descriptors that are both distinctive and insensitive to the variability we expect under changing viewpoint, illumination, and clutter. We begin at the level of the detector, which finds and links contiguous edges in scale space (§2.1). Whereas point detectors produce binary results – a feature is detected or not – the same edge in different images may be partially detected and can undergo changes in length and geometry. Our descriptor, therefore, cannot be unitary like SIFT (a fixed-length vector) but extensible, growing with the length of the associated edge. It must also be based on image statistics that are invariant to changes in illumination and insensitive to viewpoint-induced warping, making gradient histograms the natural choice. In this section, we develop the edge descriptors, including scale selection, ordering, sidedness, and discretization. How we match these descriptors is explained in §3.

2.1. Images and edge detection

We consider images to be positive-valued functions $I : \Omega \to \mathbb{R}^+; (x, y) \mapsto I(x, y)$, together with the domain $\Omega \subset \mathbb{R}^2$ where they are defined. So, when we say "image," we mean the pair $(\Omega, I) \doteq \mathcal{I}$. As edges are only meaningful at a particular scale, all computations are performed on a scale space formed by repeated convolution of the image with a Gaussian kernel. [7]

A *local edge detector* is a functional $\mathcal{D}$ that maps the image scale space onto a set of points that have certain extremal properties, for instance $\mathcal{D} : \mathcal{I} \to \{(x, y, \sigma) \mid I^\sigma_{vv}(x, y) = 0, I^\sigma_{vvv}(x, y) < 0\} \doteq \mathcal{E} \subset \mathbb{R}^3$ [7]. (Keeping with standard nomenclature, we refer to these points as *edgels.*) I^σ_v is the derivative of the image at scale σ in the direction v, and $\mathcal{E}$ is the set of points that are putatively on *some* edge, together with their intrinsic scale σ. Here, scale σ equals the variance of the Gaussian kernel for which a scale-normalized edge strength function is maximized [7]. On a contiguous subset of $\mathcal{E}$ one can define

a *linking mechanism*, formalized by a *continuous* function $\mathcal{L} : [0,T] \to \epsilon \subset \mathcal{E};\ t \mapsto \mathcal{L}(t) = (x(t), y(t), \sigma(t)) \in \epsilon$. This mechanism introduces an *order* among the edgels it links, as well as a *direction*, for in general $\mathcal{L}(t) \neq \mathcal{L}(T-t)$.

2.2. Image descriptors anchored to an edge

We denote an *image descriptor*, or *feature*, as any image statistic, or a deterministic function of an image. Note that this includes the image domain, Ω, as well as the image values $I(\Omega)$. The simplest image "descriptor" associated to an edge is the restriction of the image to the domain determined by an edge detector and particular linking:

$$\mathcal{I}_{|\mathcal{L}} \doteq \{(\bar{x}, \bar{y}, I(\bar{x}, \bar{y})) \mid (\bar{x}, \bar{y}) \in \mathcal{B}_\varsigma(x, y);\ (x, y, \varsigma) \in \epsilon\} \quad (1)$$

where $\mathcal{B}_\varsigma(x,y) = \{(\bar{x}, \bar{y}) \mid (x-\bar{x})^2 + (y-\bar{y})^2 \leq \varsigma^2\}$ is a ball of radius ς centered in (x,y). This descriptor is not very useful, as it exhibits no particular invariance properties. Rather than the raw image values of $\mathcal{I}_{|\mathcal{B}_\varsigma(x,y)}$, it is useful to employ other image statistics $\mathcal{F} : \mathcal{I} \to \mathbb{R}^K; (\Omega, I) \mapsto F(I(x,y))_{|(x,y)\in\Omega}$, which we denote in short-hand as $F(I|\Omega)$, designed to be insensitive to other nuisance factors of the image-formation process; in §2.4 we will discuss specific choices for F.

We introduce an **ordered descriptor** using the linking mechanism L as follows:

$$\phi(\mathcal{I}|\mathcal{L}) : [0,T] \to \mathbb{R}^K;\ t \mapsto F\left(I|\mathcal{B}_{\varsigma(t)}(\mathcal{L}(t))\right) \quad (2)$$

That is, $\phi(\mathcal{I}|\mathcal{L})$ is a function that maps a position on an edge to a vector based on the image region around that point. Finally, we introduce an **ordered sided descriptor** by only considering the portion of the domain $\mathcal{B}_\varsigma$ that is on one side of the edge, that is

$$\phi^+(\mathcal{I}|\mathcal{L}) : [0,T] \to \mathbb{R}^K;\ t \mapsto F\left(I|\mathcal{H}_{\varsigma^+(t)}(\mathcal{L}(t))\right) \quad (3)$$

where the "half-space" $\mathcal{H}_{\varsigma^+}$ is defined as

$$\mathcal{H}_{\varsigma^+} = \{(\bar{x}, \bar{y}) \in \mathcal{B}_\varsigma(x,y) \mid \langle (\bar{x}-x^*, \bar{y}-y^*), N(x^*, y^*)\rangle \geq 0\} \quad (4)$$

where

$$(x^*, y^*) = \arg\min_{(x,y)\in\epsilon} ||(\bar{x}, \bar{y}) - (x,y)||_2^2$$

where $N(x^*,y^*)$ is the normal vector to the edge at the point (x^*,y^*), closest to $(\bar{x}, \bar{y})$ (see figure 1). One can similarly define ϕ^- by changing the sign of the inner product with the normal.

If an edge domain intersects an **occluding boundary**, then in general neither ϕ nor $\phi^{+/-}$ are viewpoint invariant, unless the object is polyhedral. Our working hypothesis is that an ordered sided descriptor can be used in practice as an image statistic that is insensitive to viewpoint variations even in the presence of visibility artifacts, including occlusions, while at the same time being discriminative enough to enable applications to wide-baseline matching and object or category recognition. To this end, we must introduce a way to compare such descriptors. In the following sections, we explore choices of the descriptor – the actual form of the function F – as well as discuss the pratical computation, which includes discretizing the function $\mathcal{L}$.

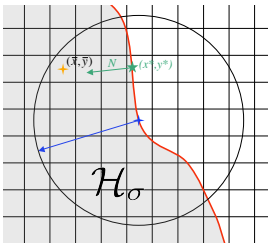


Figure 1. The domain of an edge descriptor (4). $\mathcal{H}_\sigma$ lies on one side of an edge within a circle of radius σ for a given anchor point.

2.3. Descriptor regions and discretization

Since we seek a local image descriptor for an edge, we must define its supporting domain. Edge descriptors are comprised of a list of gradient orientation histograms, each of which is computed in a domain that is anchored to the edge of interest. Choice of size and position of these domains is important if we are to retain invariance to viewpoint changes.

For practical purposes and to avoid redundancy, we compute histograms at discrete collections of points along an edge rather than on the entire continuum $[0,T]$. We call these positions *anchors*, and they will provide a canonical discretization for edge descriptors. As the positions of the anchors will influence our ability to match edges, we must select them carefully. We are fundamentally seeking image descriptors, hence the anchor positions should be tied to the image statistics and not to the geometry of the edge. [3] If an edge lies on an occluding boundary, such image statistics will change incongruously with viewpoint on either side of the edge – thus we must consider each side separately when selecting anchors. In effect, an edge has two descriptors, one for either side, each of which contains its own anchors and histograms independent of the other.

Since the histograms store the crucial matching information, we must select their support regions in a viewpoint-insensitive way. The Laplacian operator has been shown to produce scale-invariant regions for matching and is insensitive to a wide range of viewpoint transformations [8]. We adapt this scale-selection technique to find a *scale envelope* independently on both sides of an edge. For each point along an edge, we evaluate the integral of the Lapla-

[3] Regardless, since the edge geometry is not invariant under the deformations we expect, it does not make sense to find a canonical frame for our descriptor based on geometry.

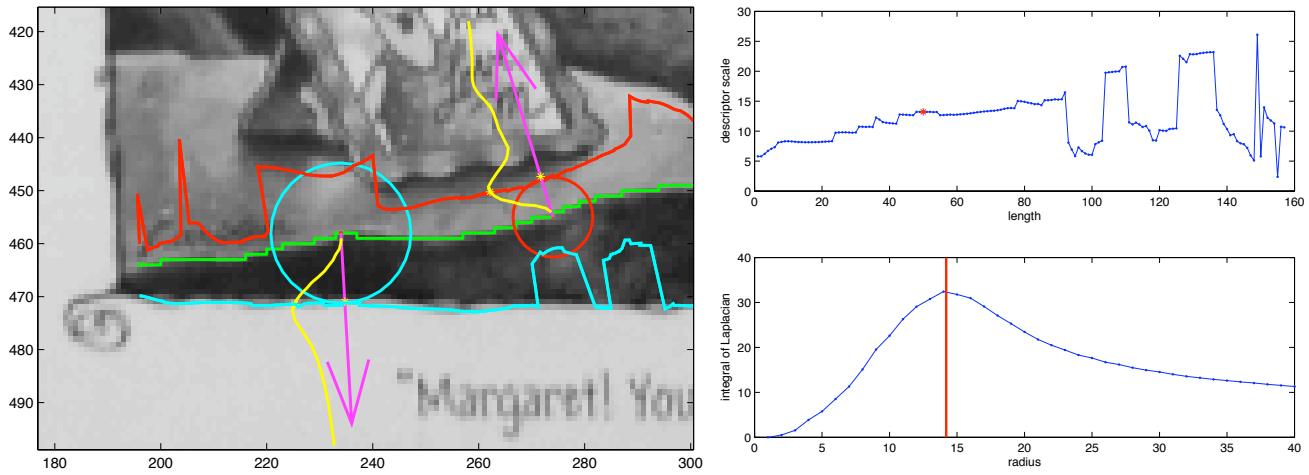


Figure 2. **Scale envelopes and descriptor scale.** (Left) The edge is plotted in green with its scale envelopes, ς^+ and ς^-, in red and cyan. The descriptor scales for two points along the edge are indicated by the red (side +) and cyan (side -) circles. The edge normals at those points are drawn as magenta arrows, along which are superimposed plots of the integral of the Laplacian vs. domain radius for that point (yellow). The selected descriptor scale (extremal Laplacian response) for each point is drawn in the direction of the edge normal at that point. (Right) Above, a plot of the lower edge envelope vs. length for the selected edge. Below, integral of the Laplacian vs. radius for the point on the edge indicated with the cyan circle.

cian within a circular region of radius r split by the edge (so that image information from only one side is used to compute the envelope for that side, as in figure 1). The r for which this integral attains an extremal value is selected as the "descriptor scale" for that point on that side. We refer to these as ς^+ and ς^- (Figure 2).

Given the scale envelopes, we now have an image-based measure for selecting anchors on either side of an edge. We choose anchors along the edge where the scale envelopes attain their extremal values. This is a reasonable choice, as although the scale envelopes may warp due to viewpoint changes, their extrema should remain at stable locations along the edge.

2.4. Gradient histograms

Of all possible F that we can use, we want to employ one that is also insensitive to other nuisance factors of the image formation process. It is well known that the gradient orientation of the intensity $I(x, y)$ is normal to the isocontours, and that the collection of level lines of an image is the maximal invariant with respect to contrast functions [1]. Therefore, it is natural to choose a statistic F based on gradient orientations. Note that gradients for the descriptor in [9] are computed at the intrinsic scale of the structure of interest detected. In our case, at the intrinsic scale computed at an edge location, the gradient orientation is constant on either side (which, essentially, defines an edge at a particular scale). Thus, the descriptor must compound information from all scales *except* the intrinsic scale σ.

Following [9], our descriptor $F\left(I|\mathcal{H}_{\varsigma^s(t)}(\mathcal{L}_i(t))\right)$ at a location $\mathcal{L}_i(t)$ on side s of edge i is comprised of an orientation histogram, binning over position and orientation, but considering all scales at points within a circle of radius proportional to ς^s on side s of the edge. As in [9], orientations are weighted by edge strength and a Gaussian centered on the anchor point. Such a histogram is computed at all anchor locations on either side (§3.1) and stored in order. This pair of ordered lists of histograms comprises the descriptor for a single edge, $\phi(\mathcal{I}|\mathcal{L}_i)$.

3. Comparing edge descriptors

Consider two images $\mathcal{I}_1$ and $\mathcal{I}_2$ that potentially portray the same scene. One can test this hypothesis by extracting invariant descriptors independently on each image and then comparing these descriptors. For each edge, we have two descriptors, ϕ^+ and ϕ^-, each of which is comprised of a list of gradient histograms. Therefore, our goal is to define a distance

$$d\left(\phi(\mathcal{I}_1|\mathcal{L}_1), \phi(\mathcal{I}_2|\mathcal{L}_2)\right). \tag{5}$$

Hidden inside $\mathcal{L}$ is a choice of endpoints, corresponding to $t = 0$ and $t = T$, as well as the parametrization: in particular, if we replace $t \in [0, T]$ with $\tau \doteq h(t)$ with h being any continuous and monotonic function such that $h(0) = 0$, $h(T) = T$, we obtain a different value of $\phi(\mathcal{I}|\mathcal{L} \circ h)$ for each different h even though the underlying descriptor is the same. In other words, h is a *nuisance parameter* that must be factored out of the comparison. This can be done in two ways: one is to choose amongst all possible parametrizations h one that is *canonical*, in the sense

of being tied to the data and chosen independently in each image. The other is to write d above as a function of h_1, h_2, and then define the distance as the minimum over h_1, h_2. In practice, we use a combination of both for matching edge descriptors across images.

3.1. Canonical edge descriptors

In order to make the descriptor ϕ independent of the parametrization h, we can choose a *canonical parametrization*, $\hat{h}$, in a way that is dependent only on the data (without requiring comparison between images) and repeatable across viewpoints. One strategy, outlined in §2.3, consists of selecting critical points on the time segment $t \in [0, T]$, for instance all extrema of $\varsigma^s(t)$, and then choose the warping $\hat{h}$ that brings such points into fixed positions. For instance, if $t_1, \ldots, t_N$ are N critical points in the interval $[0, T]$, then we can select

$$\hat{h} : [0, T] \to [0, T] \mid \hat{h}(t_i) = T(i-1)/(N-1),\ i = 1 \ldots N \quad (6)$$

with $t_1 = 0$ and $t_N = T$. Given this canonical warping, we could simply compare the lists of histograms directly, assuming all critical points of a particular edge are present in all possible viewpoints. Unfortunately, this is typically not the case due to the effects of clipping and rasterization. The following sections detail our solution to this problem.

3.2. Matching edges with dynamic programming

The canonization procedure above allows us to define a distance that can be computed in closed-form without the need to solve costly optimization procedures as part of the comparison. However, the choice of canonical element $\hat{h}$ is fragile, in that extrema of $\mathcal{L}$ can appear or disappear due to noise, rasterization effects from viewpoint-induced warping, or incorrect edge linking on a particular image. Furthermore, despite fixing a canonical parameterization for each edge, there are still a number of effects we need to account for in the process of matching. These are: (a) differing starting and ending points caused by clipping of an edge, (b) inconsistent polarity, meaning the starting and ending points of the edge are reversed, (c) loss of anchor locations in one or both edges, and (d) matching one or both sides of the edge descriptors (handling occluding boundaries).

A solution is to choose the optimal $\hat{h}_i$ as part of the matching process by defining $d\left(\phi(\mathcal{I}_1|\mathcal{L}_1), \phi(\mathcal{I}_2|\mathcal{L}_2)\right)$ as the minimum

$$\min_{h \in \mathcal{M}} \| F\left(I_1 | \mathcal{B}_{\varsigma_1^s(t)}(\mathcal{L}_1(t))\right) - F\left(I_2 | \mathcal{B}_{\varsigma_2^s(h(t))}(\mathcal{L}_2(h(t)))\right) \|^2$$

constrained to h belonging to the set of monotonic diffeomorphisms with fixed boundary conditions. A discrete version of this problem can be solved using Dynamic Programming, and is known as "dynamic time warping." [20] Dynamic time warping (DTW) finds the optimal alignment between two discrete signals, sampled on differing grids, accounting for missing parts and differing lengths.

DTW handles problems (a) and (c) automatically, as the warping function allows differing endpoints and missing anchors. The effects of reversed polarity and separate matching of the sides can be dealt with by simply calculating the minimum of four path costs, one for each unique combination of reversal and sidedness. Note that there are only four due to symmetry: reversing only one edge before DTW yields the same cost as reversing only the other, and reversing both is equivalent to no reversal at all. Since we assign sides based on the normal vector along the edge, we know that when computing DTW assuming the same polarity, we have only to compare ϕ_1^+ with ϕ_2^+ and ϕ_1^- with ϕ_2^-, while opposing sides are compared when assuming a reversal.

3.3. Smith-Waterman matching

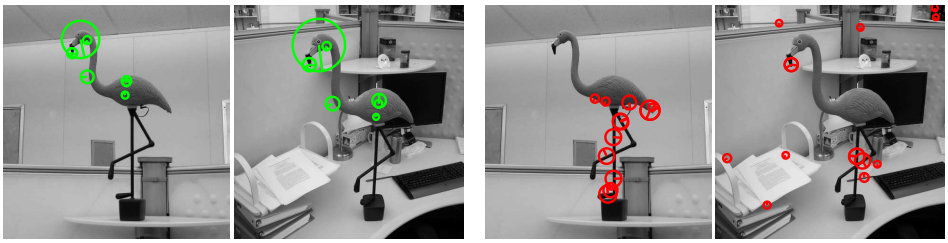
While DTW can be applied to this problem in general, we have found it more effective and efficient to utilize a special case of DTW, most often used for protein sequence alignment. The Smith-Waterman (SW) algorithm [17] is used to find alignments in protein sequences allowing for gaps, differing lengths, and errors. This is a natural analog to our problem of edge descriptor matching, as geneticists often need to match subsequences of proteins that include gaps, errors, and mismatches. Unlike DTW, SW relies on a fixed alphabet of possible "letters" in a sequence and a defined matching score for all possible pairs of letters, including a penalty for matching a letter to a gap.

In order to apply the SW algorithm to matching edge descriptors, we must first transform our histograms into letters in a fixed alphabet. We cluster all the histograms in an image using K-means, assigning each a cluster center (each cluster center is itself a histogram). The edge descriptors are then transformed from lists of histograms to lists of letters, one letter per cluster center. The Euclidean distance between any pair of centers is used to define the matching score for a letter pairing.

Like DTW, SW returns both an *alignment* and matching *score* for the edge descriptors being compared. This allows us to find matching edges as well as matching segments, which is important since edges can undergo clipping between viewpoints.

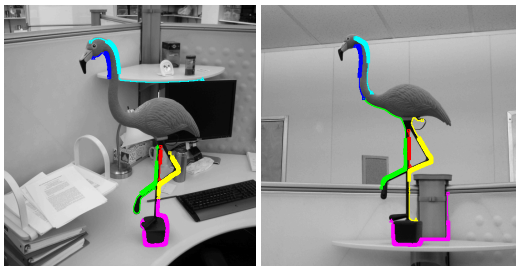
4. Experiments

The following experiments demonstrate the utility and effectiveness of edge descriptors for a variety of tasks. These include the classic correspondence problem of matching edges in image pairs portraying the same scene or object, structure from motion (SFM) from line segments, and object category recognition.



(a) Correct SIFT matches on the flamingo.

(b) Erroneous SIFT matches on the flamingo. Most have domains which cross an occluding boundary.



(c) Edge matches on the flamingo.

Figure 3. **Occlusions and clutter.** The object of interest, a lawn flamingo, is placed in front of two different backgrounds. This is detrimental to SIFT or similar point-feature descriptors, since the descriptor domain cannot account for occluding boundaries.

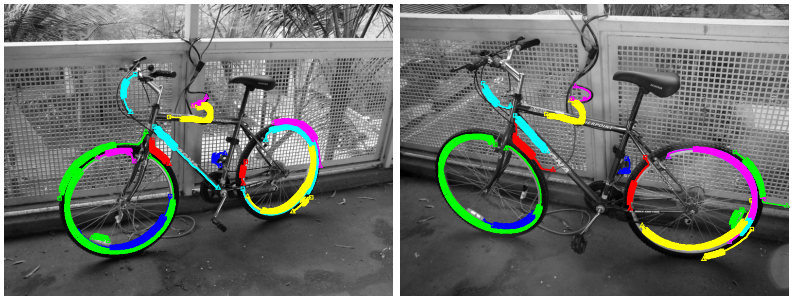


Figure 4. **Change of viewpoint.** Matching bicycle edges under viewpoint changes. There is some unavoidable confusion between the front and rear wheels since no global geometric constrains are enforced in the matching.

4.1. Correspondence

As edge matching is our primary goal, we have extensively tested our system on a variety of scenes that contain edge features. Some examples are provided here of scenes in which edge matching is successful despite changing viewpoint, clutter, variable background (an effect of occlusion). In all figures in this section (3,4), like colors indicate edge matches between the image pair (though the colors are repeated, so it is not a one-to-one relationship). Additionally, since we can find matching portions of edges, these matched segments are displayed as a bold overlay on top of the entire connected edge.

4.2. Straight-line structure from motion

To demonstrate a direct application of our edge matching system, we implemented a front-end to C.J. Taylor's 1994 algorithm for finding the structure and motion of a scene from straight line segments [19]. In the original paper and all subsequent work, edges are selected using a Canny edge detector, then straight-line segments are matched by hand across many views. [19] models a scene as a collection of straight 3D line segments whose projections appear in the input images. By minimizing the reprojection error between these model lines and the data edges, parameters of the 3D lines can be found along with the camera motion between frames.

The original code and data Taylor used in his experiments are available on the web ([18]), and we have demonstrated the ability to automate these experiments using our edge matcher as a front-end. First, edges are extracted and descriptors generated on each data image, following which these are matched against the first view, all according to §2.1. Because the SFM algorithm requires straight edges, we decompose these matched edges into straight segments and use our knowledge of edge alignments to obtain correspondence between these segments. We first match then decompose because matching longer edges is more robust than short segments. In any automatic matching procedure, mismatches are inevitable. The original SFM procedure is not robust to errors in correspondence (as matches are cho-

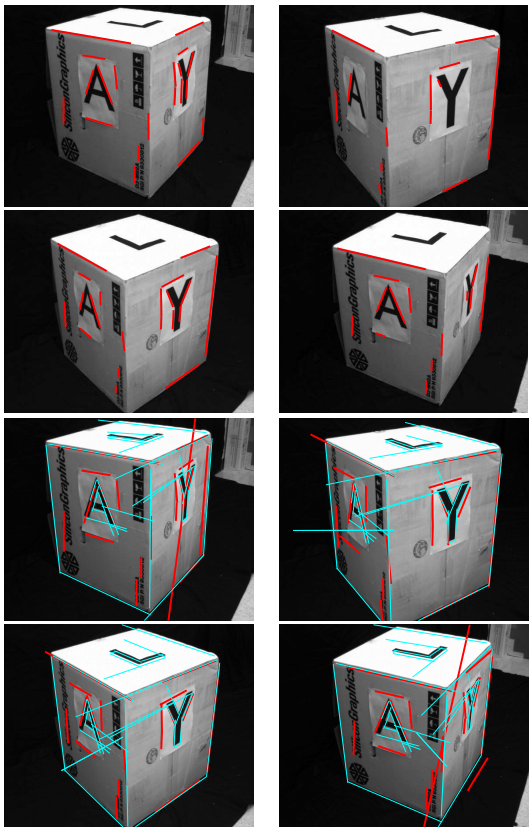


Figure 5. (Top four) Straight-line inlier segments used for SFM shown in red. (Bottom four) Reprojections of edge segments onto the original images. Light blue lines represent the reprojections of edges estimated using Taylor's original data. Heavy red lines are the reprojections using automatically matched edge segments.

sen by hand), so we resort to using a RANSAC procedure to pick inlier matches while finding structure and motion. An alternative approach would be to make the optimizations in [19] robust to mismatches by using robust statistical techniques, such as M-estimators.

Our experiments show performance nearly as good as Taylor's original experiments using only line segments extracted and matched automatically (Figures 5, 6). The final estimates are somewhat degraded in comparison to the original since the edge matching system cannot match all of the small edges in the scene and must discard ambiguous matches, whereas the original experimental data is matched by hand and includes the maximum number of edge segments.

4.3. Category recognition

Another application which may benefit from edge descriptors is category recognition. This topic has been heavily researched in recent years with many data sets and techniques emerging. The Caltech 101 data set [4], which includes 101 categories of objects plus a background category, has become one of the benchmarks for testing recognition algorithms. To determine whether or not performance in this task can be improved by the inclusion of edge descriptors, we evaluate two recognition systems on Caltech 101 with the inclusion of edge descriptors and without. Both [14] and [5] are "bag of features" techniques, the former being the most basic version and the latter including a spatial model which improves performance. In the simplest terms, these consider images as collections of visual "words," which are formed using hierarchical K-means clustering of SIFT descriptors (weighted histograms of gradient orientations). These are used to train a classifier to discriminate between categories.

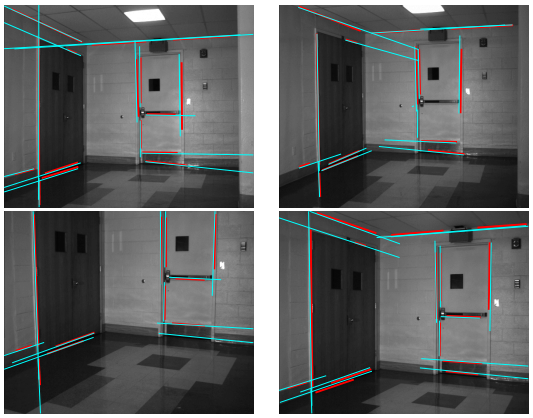


Figure 6. Reprojections of edge segments onto the original images of the hallway sequence. Light blue lines represent the reprojections of edges estimated using our edge features, and heavy red lines show our edge features used for SFM. The reprojections align very closely with the original straight-line segments the system selected as inliers.

Our methodology in this case differs from previous experiments. Because the bag of features technique finds prototype feature histograms, the simple algorithms that we test assume that a single type of descriptor is extracted from the data. At root, edge descriptors are lists of orientation histograms very similar to SIFT descriptors, except they are computed over multiple scales on a domain that covers only one side of an edge. To test the most basic scheme for adding these features to the bag, we find edge descriptors as usual but do not consider them in a chain. Rather, we sample the histograms that are developed along each edge and treat them as individual features. This makes the addition of edge descriptors transparent to the classifier.

Even using this simple scheme to add our new features, we find a significant boost in recognition performance. Table 1 summarizes our results. We perform two sets of experiments, one using the basic bag of features from [14] and one with the spatial pyramid matching technique of [5].

For each, the classifier is given either SIFT features alone or SIFT features plus edge descriptors; all other parameters of the classifier remain identical between trials. Results are reported as confusion matrices whose entries indicate the proportion of times images from a class i are classified as class j; rows and columns sum to 1. The ideal result is the $N \times N$ identity matrix, where N is equal to the number of classes. For clarity of presentation, we show only the mean and median of the diagonal of a confusion matrix, indicating the overall frequency of correct classifications.

The addition of edge descriptors boosts average performance on Caltech 101 using both classification techniques. We hypothesize that the accurate detection of edges and descriptor splitting on either side provides additional robustness to the effects of occlusions, as illustrated in Figure 3. Standard SIFT descriptors encompass an entire circular domain, whether or not this domain crosses the boundary between foreground objects and background. This is mitigated by incorporating edge descriptors, as some of the features given to the classifier will be robust to this effect. We anticipate further gains in performance when we incorporate a more systematic method of categorization that includes the edge descriptor chaining mechanism.

Mean confusion			
	SIFT only	**SIFT+Edges**	Difference
Bag	0.4246	**0.4999**	0.0753
SPM	0.5434	**0.6253**	0.0819
Median confusion			
Bag	0.3515	**0.4479**	0.0964
SPM	0.5300	**0.6214**	0.0914

Table 1. Mean and median performance (mean or median of diagonal of confusion matrix) for each technique using SIFT only, SIFT+Edges, and the differential.

5. Conclusions

We have presented a method for matching edges in different images of the same scene undergoing wide-baseline viewpoint changes. By taking advantage of edge ordering and sidedness, these descriptors are robust to the effects of occlusion and edge clipping, as well as illumination and affine deformations. Our results demonstrate the effectiveness of the technique for a variety of tasks, including correspondence, SFM, and category recognition. We anticipate that continued research will yield more benefits for category and object recognition.

6. Acknowledgements

This research was supported by AFOSR grant FA9550-06-1-0138, NSF ECS-0622245, and ONR 67F-1080868. The authors would also like to thank Brian Fulkerson and Andrea Vedaldi for sharing code and ideas.

References

[1] L. Alvarez, F. Guichard, P. L. Lions, and J. M. Morel. Axioms and fundamental equations of image processing. *Arch. Rational Mechanics*, 123, 1993.

[2] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8:679–698, 1986.

[3] A. J. Davison et al. Monoslam: Real-time single camera slam. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6), 2007.

[4] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *IEEE CVPR Workshop of Generative Model Based Vision*, 2004.

[5] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bag of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recogn.*, 2006.

[6] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1465–1479, 2006.

[7] T. Lindeberg. Edge detection and ridge detection with automatic scale selection. *Int. J. of Computer Vision*, 30(2):77–116, 1998.

[8] T. Lindeberg. Feature detection with automatic scale selection. *IJCV*, 30(2):77–116, 1998.

[9] D. G. Lowe. Distincive image features from scale-invariant keypoints. *Int. J. of Computer Vision*, 2(60):91–110, 2004.

[10] J. Malik, S. Belongie, T. Leung, and S. Jianbo. Contour and texture analysis for image segmentation. *Int. Journal of Computer Vision*, 43:7–27, 2001.

[11] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. Technical report, CTU Prague and University of Sidney, Center for Machine Perception and CVSSP.

[12] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proceedings of the 7th European Conference on Computer Vision*, pages 128–142. Springer-Verlag, 2002.

[13] K. Mikolajczyk, A. Zisserman, and C. Schmid. Shape recognition with edge-based features. In *British Machine Vision Conference*, volume 2, pages 779–788, September 2003.

[14] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recogn.*, pages 2161–2168, Washington, DC, USA, 2006. IEEE Computer Society.

[15] A. R. Pope and D. G. Lowe. Learning appearance models for object recognition. In J. Ponce, A. Zisserman, and M. Hebert, editors, *Object Representation in Computer Vision II*, volume 2, pages 201–219, 1996.

[16] M. A. Ruzon and C. Tomasi. Corner detection in textured color images. In *Proc. of Intl. Conf. on Computer Vision*, 1999.

[17] T. Smith and M. Waterman. Identification of common molecular sequences. *J. Molecular Biology*, 147:195–197, 1981.

[18] C. J. Taylor. `http://www.cis.upenn.edu/~cjtaylor/projects/YaleSFM/software_files/software.html`.

[19] C. J. Taylor and D. J. Kriegman. Structure and motion from line segments in multiple images. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 17(11):1021–1032, 1995.

[20] J. Treichler and B. Agee. A new approach to multipath correction of constant modulus signals. *IEEE Trans. Acous. Speech Signal Process.*, 31(2):459–472, 1983.

[21] Y. Tsin, Y. Genc, Y. Zhu, and V. Ramesh. Learn to track edges. In *Proc. of Intl. Conf. on Computer Vision*, 2007.

[22] T. Tuytelaars and L. V. Gool. Wide baseline stero matching based on local, invariant regions. Technical report, University of Leuven.

[23] A. Vedaldi and S. Soatto. Features for recognition: viewpoint invariance for non-planar scenes. In *Technical Report UCLA-CSD 04-0049*, December 2004.