# Discriminative Image Warping with Attribute Flow

Weiyu Zhang, Praveen Srinivasan
GRASP Laboratory, University of Pennsylvania
3330 Walnut St., Philadelphia, PA - 19104
{zhweiyu, psrin}@seas.upenn.edu

Jianbo Shi
GRASP Laboratory, University of Pennsylvania
3330 Walnut St., Philadelphia, PA - 19104
jshi@cis.upenn.edu

## Abstract

*We address the problem of finding deformation between two images for the purpose of recognizing objects. The challenge is that discriminative features are often transformation-variant (e.g. histogram of oriented gradients, texture), while transformation-invariant features (e.g. intensity, color) are often not discriminative. We introduce the concept of attribute flow which explicitly models how image attributes vary with its deformation. We develop a non-parametric method to approximate this using histogram matching, which can be solved efficiently using linear programming. Our method produces dense correspondence between images, and utilizes discriminative, transformation-variant features for simultaneous detection and alignment. Experiments on ETHZ shape categories dataset show that we can accurately recognize highly deformable objects with few training examples.*

## 1. Introduction

Consider two images $\mathcal{I}$ and $\mathcal{J}$, we are interested in finding deformation and correspondence between them for the purpose of recognition. In particular, we consider image $\mathcal{I}$ to be a single model of certain object category. Our goal is to detect the object from the same category in image $\mathcal{J}$, and align the object model to the detection. Both detection and alignment are done in a one-shot fashion.

The key concept we propose is to model image deformation as a flow of image attributes, shown in Figure 1 & 2. Instead of modeling *2D spatial transformation* of pixels or feature points, we model *attribute transformation* on image features, such as edge orientation or histogram of oriented image gradient, in a higher dimensional space.

Our formulation has the advantages that it can use a broad set of image features that leads to more robust and faster alignment. For example, when an elongated object rotates, its edge orientation histogram shifts in the angular bins. Computing this shift is much easier for such objects than searching over all possible rotations of the object.

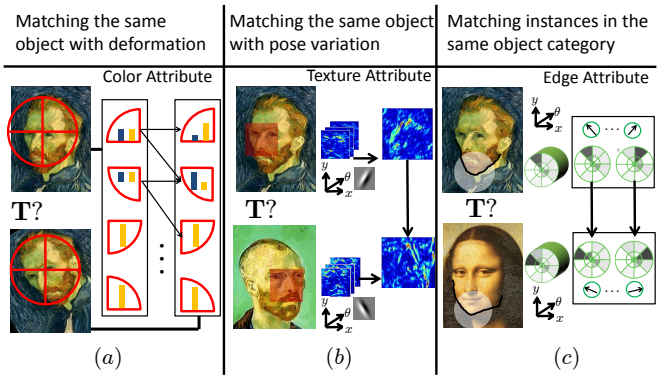


Figure 1. Flow in attribute space leads to image matching under deformation. (a) uses flow in color histogram for matching rotated images. (b) uses flow in orientation of image gradient for matching objects with different pose. (c) uses flow in oriented edge shape context for matching object instances in the same category.

To compute a *attribute transformation* there are three questions to be resolved:

1. how to constraint *attribute transformation* such that it finds a solution that has a valid *2D spatial transformation*? In general, attribute transformation has more degrees of freedom than spatial transformation. In our example, the two adjacent angular bins can shift in opposite directions, leading to an inconsistent spatial transformation. To prevent such case, we need to map constraints in the spatial transformation space into constraints on the *attribute transformation*. This is a key part of our algorithm, we derive its solution in detail in sections below.

2. how to compute *attribute transformation* efficiently? As in our example, we will use histogram to represent the continuous attribute in a discrete non-parametric form. Finding a flow in the histogram space amounts to computing an optimal bipartite graph matching. This can be solved efficiently using linear programming. Again, we need to ensure that the geometrical

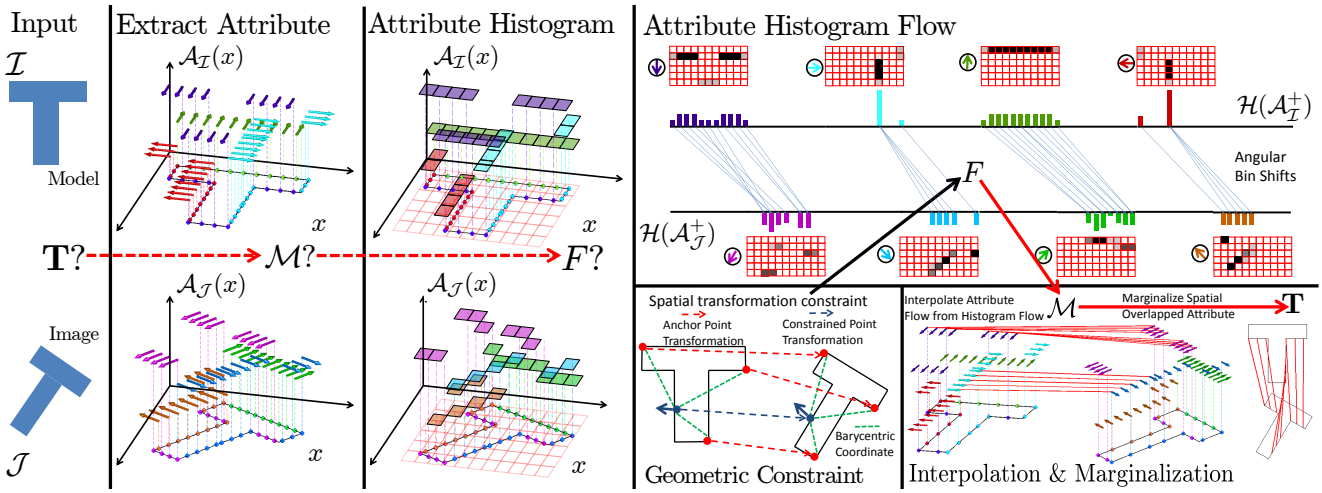


Figure 2. **Work flow of attribute flow computation.** Given an input of a model image $\mathcal{I}$ and a test image $\mathcal{J}$, we seek an optimal spatial transformation ($\mathbf{T}$) for alignment. We first extract oriented edge attribute (arrows) of both $\mathcal{I}$ and $\mathcal{J}$ in the "Extract Attribute" step. The color and orientation of the arrows represent the edge normal orientation. In the "Attribute Histogram" step, we quantize spatial location and edge orientation into histogram bins. We visualize attribute histogram space as a 3D volume, where an image is divided into a spatial grid, and each grid cell carries a stack of bins representing edge orientation histogram at that location. The color of each bin indicates its edge orientation. In "Attribute Histogram Flow" step we compute the optimal histogram flow ($F$) (blue lines) to match the attribute histogram of $\mathcal{I}$ and $\mathcal{J}$. Red grids and colored bars show the histogram counts in image space and "flattened" histogram of oriented edge respectively. We impose "Geometrical Constraints" when solving $F$. For affine transformation, given transformation on anchor points, transformation on another point is constrained by preserving the barycentric coordinate. In the "Interpolation and Marginalization" step, we compute attribute flow ($\mathcal{M}$) from $F$, and compute spatial transformation $\mathbf{T}$ by marginalizing $\mathcal{M}$ over spatially overlapped attributes.

constraints on the *attribute transformation* are passed down to the computation of the histogram flow. Once we computed flow in the histogram space, we interpolate to obtain the continuous *attribute transformation*.

3. how to map a *attribute transformation* in the feature space back to a *spatial transformation*? In our example, once we have computed how the histogram bins are shifted, multiplying the shift (in bins) by the angular bin width gives the angular rotation. We will show how to combine the *attribute transformation* with the attribute features to compute the *spatial transformation* through a process of marginalization.

We formulate the *attribute transformation* as *Attribute Flow*, addressing the three questions above. Figure 2 shows the key steps of computing attribute flow.

We further extend this formulation by also allowing for reasoning over which image regions contribute to the attribute feature space. For example, when detecting object in cluttered environment, it is important to only allow image contours that are actually part of the true object boundary to participate in the matching, an idea first introduced by Zhu et al. ([14]).

The paper is organized as following. We show in Sec. 2 how attribute flow is formulated, and how affine constraints on spatial transformations are mapped into attribute flow. We show in Sec. 3 how efficient computation of attribute flow is achieved using histogram flow formulation, and how constraints are passed from attribute flow to histogram flow. In Sec. 4, we show how to select the correct image regions for matching under clutter. In Sec. 6 we demonstrate our method on the ETHZ Shape Classes Dataset [1].

## 2. Image alignment through Attribute Flow

### 2.1. Attribute Flow

We define attribute of image $\mathcal{I}$ using a vector-valued function $\mathcal{A}_\mathcal{I} : \mathbb{R}^2 \rightarrow \mathbb{R}^n$, which maps each image location $x$ to an $n$-dimensional attribute vector $\mathcal{A}_\mathcal{I}(x)$ computed from its surrounding information. The attributes of the entire image $\mathcal{I}$ form a vector field. $\mathcal{A}_\mathcal{J}$ denotes the attribute function of image $\mathcal{J}$.

Let $\mathbf{T} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ be the optical flow that represents the image deformation. Under desired $\mathbf{T}$ between two images, we expect the attribute $\mathcal{A}_\mathcal{J}(\mathbf{T}(x))$ of deformed image $\mathcal{J}$, and attribute $\mathcal{A}_\mathcal{I}(x)$ of image $\mathcal{I}$ to be similar. This amounts to finding $\mathbf{T}$ that minimizes the attribute dissimilarities at corresponding locations, as defined in the following **Attributed Optical Flow** problem:

$$\min_{\mathbf{T}} \int_{\mathbb{R}^2} ||\mathcal{A}_\mathcal{I}(x) - \mathcal{A}_\mathcal{J}(\mathbf{T}(x))||_p dx \qquad (1)$$

Optimizing over $\mathbf{T}$ in the above equation requires synthesizing the transformed image attributes for different hypotheses of $\mathbf{T}$. The process of extracting attributes from a transformed image can be highly non-linear, making direct optimization of (Eq.1) difficult.

Instead, we seek an explicit explanation of how attributes are mapped across two images. We define generalized image attribute $\mathcal{A}^+_{\mathcal{I}}(x) = (x, \mathcal{A}_{\mathcal{I}}(x)) \in \mathbb{R}^{n+2}$ in the (generalized) attribute space $\mathbb{R}^{n+2}$.

Let attribute flow be the mapping $\mathcal{M} : \mathbb{R}^{n+2} \to \mathbb{R}^{n+2}$ between attribute spaces of image $\mathcal{I}$ and image $\mathcal{J}$. The mapping $\mathcal{M}(\mathcal{A}^+_{\mathcal{I}}(x)) = (\mathcal{M}_1(x), \mathcal{M}_2(\mathcal{A}^+_{\mathcal{I}}(x)))$ is defined such that $\mathcal{M}_1(x) = \mathbf{T}(x)$ is the optical flow. We seek an **Attribute Flow** $\mathcal{M}$ that minimizes:

$$\min_{\mathcal{M}} \int_{\mathbb{R}^{n+2}} |\delta(y - \mathcal{M}(\mathcal{A}^+_{\mathcal{I}})) - \delta(y - \mathcal{A}^+_{\mathcal{J}})|_1 dy \quad (2)$$

where $\delta(y)$ is Dirac delta function in attribute space $\mathbb{R}^{n+2}$.

An attribute flow needs to have a meaningful geometric interpretation. In this paper, we show how to impose affine transformation constraint on the attribute flow, when image attributes are orientations of edges or image gradients.

### 2.2. Affine Constraint on Spatial Transformation $\mathbf{T}$

Three non-collinear points and their optical flow uniquely determine an affine transformation. We call these three points anchor points and denote as $x_1, x_2, x_3$. We can represent any other point $x_p$ using the affine combination of them: $x_p = \sum_{i=1}^{3} \alpha_i^p x_i$, where $(\alpha_1^p, \alpha_2^p, \alpha_3^p)$ is also known as the barycentric coordinate. Barycentric coordinate stays invariant under affine transformation:

$$\mathbf{T}(x_p) = \sum_{i=1}^{3} \alpha_i^p \mathbf{T}(x_i) \quad (3)$$

For oriented features, when the image rotates, their orientations should change accordingly. We want to link the orientation change with the anchor points transformation. For any point $x_p$ with orientation $\theta_p$, let $(\beta_1^p, \beta_2^p, \beta_3^p)$ be the barycentric coordinate of an imaginary points $x'_p = x_p + r(\theta_p)$, where $r(\theta_p)$ is the unit vector along direction $\theta_p$:

$$r(\theta_p) = [\cos(\theta_p), \sin(\theta_p)]^\mathsf{T} \in \mathbb{R}^2$$

Since the barycentric coordinates of both $x_p$ and $x'_p$ stay invariant under affine transformation, we have

$$r(\mathbf{T}(x_p, \theta_p)) \sim \sum_{i=1}^{3} (\beta_i^p - \alpha_i^p) \mathbf{T}(x_i) \quad (4)$$

where, with a slight abuse of notation, $\mathbf{T}(x_p, \theta_p)$ represents the transformed orientation of $\theta_p$ at location $x_p$. $r(\mathbf{T}(x_p, \theta_p))$ is the unit vector along direction $\mathbf{T}(x_p, \theta_p)$. The symbol $\sim$ represents vector similarity up to a scale difference.

### 2.3. Affine Spatial Transform Constraint on Attribute Flow $\mathcal{M}$

Considering orientation as the image attribute, the attribute flow $\mathcal{M}$ is defined on each image location $x_p$ and its orientation $\theta_p$:

$$\mathcal{M}(x_p, \theta_p) = (\mathbf{T}(x_p), \mathbf{T}(x_p, \theta_p))$$

To recover the image transformation from $\mathcal{M}$ we index the location attribute as $\mathbf{T}(x_p) = \mathcal{M}_1(x_p)$. To get the transformed orientation at each location, we index both location and orientation attributes as $\mathbf{T}(x_p, \theta_p) = \mathcal{M}_2(x_p, \theta_p)$. Adapting affine constraints on optical flow $\mathbf{T}$, we have the **Constrained Attribute Flow** problem as:

$$\begin{aligned} \min_{\mathcal{M}} \quad & \int_{x,\theta} |\delta([x,\theta] - \mathcal{M}(\mathcal{A}^+_{\mathcal{I}})) - \delta([x,\theta] - \mathcal{A}^+_{\mathcal{J}})|_1 dx d\theta \\ \text{s.t.} \quad & \forall p, \quad \mathcal{M}_1(x_p) = \sum_{i=1}^{3} \alpha_i^p \mathcal{M}_1(x_i) \\ & r(\mathcal{M}_2(x_p, \theta_p)) \sim \sum_{i=1}^{3} (\beta_i^p - \alpha_i^p) \mathcal{M}_1(x_i) \end{aligned} \quad (5)$$

We seek an efficient optimization of $\mathcal{M}$ in (Eq. 5) in the following section using histogram.

## 3. Histogram Flow $F$

We represent the attributes $\mathcal{A}^+$ in a non-parametric form using an attributed histogram function $\mathcal{H} : \mathbb{R}^{n+2} \to \mathbb{Z}^m$. This function maps the image attributes to a set of counts for $m$ different histogram bins. $\mathcal{H}(\mathcal{A}^+_{\mathcal{I}})$ and $\mathcal{H}(\mathcal{A}^+_{\mathcal{J}})$ denote the attribute histogram of image $\mathcal{I}$ and image $\mathcal{J}$ respectively.

Recall the first component of $\mathcal{A}^+$ is spatial location and the second component is orientation. If we quantize location into $m_x$ cells and orientation into $m_\theta$ bins, we have total of $m = m_x \times m_\theta$ attribute bins. We can visualize these bins as a 3D bin space with a stack of $m_\theta$ bins on each of $m_x$ quantized 2D spatial locations.

Each bin $k = 1, \ldots, m$ has associated with it an instantiation of the attribute: $(x_k, \theta_k)$, where $x_k$ represents the spatial center location of bin $k$ and $\theta_k$ represents the dominant orientation of bin $k$. The histogram value in each bin indicates how many image pixels/edges (could be fractional) are mapped into the quantized bin attribute.

Between the histogram bins in the two images, we define the *normalized histogram flow*: $F \in \mathbb{R}^{m \times m}$, where $F_{k,l}$ measures the normalized volume of the contents flowing from bin $k$ in image $\mathcal{I}$ to bin $l$ in image $\mathcal{J}$.

The normalized flow should obey conservation constraint: the outflow of each bin $k$ must sum up to 1. As

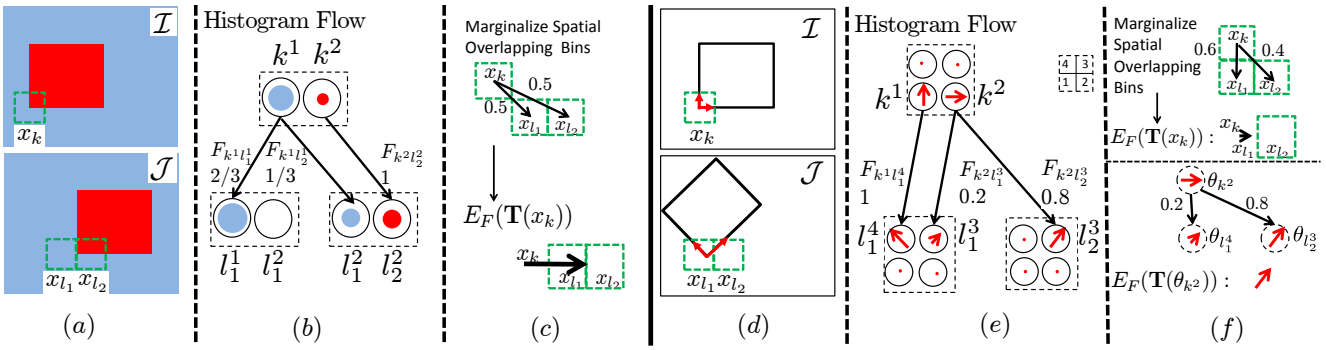


Figure 3. **Estimate local translation using color histogram**: (a) shows two images containing translated red square in blue background superimposed by image grids (dashed square). (b) shows color histograms on image grids and normalized histogram flow $F$ between them. Each grid location (dashed square) has two color histogram bins (solid circle) indexed by the superscript of the bin labels. The areas of color dots in each bin represents the histogram counts. (c) calculates the expected local translation of $x_k$ by marginalizing (weighted by model feature) over color histogram bins at the same grid location. **Estimate local translation and rotation using oriented edge histogram**: (d) shows edge map of two images containing translated and rotated square superimposed by image grids (dashed square). (e) shows orientated edge histogram on image grids and normalized histogram flow $F$. Each grid location (dashed square) has four edge orientation bins (solid circle) indexed by the superscript of the bin labels. The direction and length of arrow inside each bin represent the edge orientation and oriented edge counts respectively. (f) calculates the expected local translation of $x_k$ (top), and expected local rotation of bin $k^2$ (bottom). Again we need to marginalize over spatially overlapping bins to calculate local translation.

a non-parametric estimate of attribute flow, we expect **Attribute Histogram Flow** to minimize the following cost function:

$$\begin{aligned} \min_{F} \quad & |F^{\mathsf{T}}\mathcal{H}(\mathcal{A}^{+}_{\mathcal{I}}) - \mathcal{H}(\mathcal{A}^{+}_{\mathcal{J}})|_1 \\ \text{s.t.} \quad & F\mathbf{1} = \mathbf{1} \quad F \geq 0 \end{aligned} \tag{6}$$

We can visualize histogram flow as moving histogram mass across the two histogram space. Image often contains salient structures in the histogram space (e.g. object with a dominate edge orientation, or a unique color histogram), which facilitates the matching between attribute histograms.

We do not want histogram mass to move independently across the bins, as it might lead to inconsistent geometrical transformation. In the following, we show how to impose affine spatial transformation constraint on the attribute histogram flow computation.

### 3.1. Spatial Transformation from Histogram Flow

Normalized histogram flow $F$ can also be viewed as a probability encoding of the quantized attribute flow $\mathcal{M}$: if $F_{k,l}$ is large, then we expect that the attribute flow $\mathcal{M}$ should more likely map attribute $(x_k, \theta_k)$ to $(x_l, \theta_l)$.

When normalized histogram flow $F_{k,l} = 1$, we can use $\mathbf{T}(x_k) = F_{k,l}x_l = x_l$ to encode $x_k$ in image $\mathcal{I}$ has moved to $x_l$ in image $\mathcal{J}$. In general, given $F_{k,l}$ for all possible $l$, we estimate the *expected* spatial translation $E_F(\mathbf{T}(x_k))$ under this probability as

$$E_F(\mathbf{T}(x_k)) = \sum_l F_{k,l}x_l \tag{7}$$

When there are spatially overlapping bins, we can obtain the expected translation $\mathbf{T}(x_k)$ by further marginalizing over associated bins.

Since each histogram bin $k$ can index into unique location and rotation, we use $\mathbf{T}(\theta_k)$ instead of $\mathbf{T}(x_k, \theta_k)$ to represent the transformed orientation of bin $k$. Its expectation can be written as follows:

$$E_F(\mathbf{T}(\theta_k)) = \sum_l F_{k,l}\theta_l \tag{8}$$

Figure 3 illustrates the process of computing expected local translation and rotation of histogram bins using color histogram and oriented edge histogram. To get the continuous attribute flow $\mathcal{M}$, we can do interpolation using the discrete expected local translations and rotations on histogram bins.

### 3.2. Affine Spatial Transformation Constraints in Histogram Flow

Affine constraints for the attribute flow $\mathcal{M}$ needs to be passed onto the computation of histogram flow $F$. Given three non-collinear anchoring bins $k_1, k_2, k_3$, ideally we have $E_F(\mathbf{T}(x_k)) = \sum_{i=1}^{3} \alpha_i^k E_F(\mathbf{T}(x_{k_i}))$ for arbitrary bin $k$, where $\alpha_i^k$ is the barycentric coordinate of the bin spatial attribute $x_k$. Due to the quantization error introduced by histogram function, we can instead express this as a penalty for two quantities being different, which we call

$$\text{AffCon}_{\mathbf{x}}(F,k) : |E_F(\mathbf{T}(x_k)) - \sum_{i=1}^{3} \alpha_i^k E_F(\mathbf{T}(x_{k_i}))|_1 \tag{9}$$

For the orientation attribute $\theta_k$ of bin $k$, we adapt the following constraint from (Eq.4) for transformed orientation:

$$r(E_F(\mathbf{T}(\theta_k))) \sim \sum_{i=1}^{3}(\beta_i^k - \alpha_i^k)E_F(\mathbf{T}(x_{k_i}))$$

Because the relationship is only proportionality and not equality, the method of encoding an equality constraint softly by computing the distance between the two quantities does not apply. Instead, we must introduce a scaling factor $s$ that compensates for the proportional relationships between the two. We define this error function as following:

$$\min_s |s\cdot r(E_F(\mathbf{T}(\theta_k)))-\sum_{i=1}^{3}(\beta_i^k-\alpha_i^k)E_F(\mathbf{T}(x_{k_i}))|_1 \quad (10)$$

Instead of minimizing the function over $s$, we introduce augmented histogram flows $G^c$ and $G^s$, which encode both histogram flow $F$ and quantized scaling factors. For each entry $F_{k,l}$ of original flow $F$, there are several copies in the augmented flows $G^c, G^s$. Each copy, denoted as $G^c_{k,l,q}$ ($G^s_{k,l,q}$), defines the flow between histograms on quantized cos (sin) value of orientation $\theta_k$ and $\theta_l$, with quantized scale change $s_q$. Thus above affine constraint can be approximated by the following linear cost function with the marginalized constraint of $G^c$ and $G^s$, which we call

$$\begin{aligned}&\text{AffCon}_\theta(F,k):\\ &\min_G \quad |[E_{G^c}(s_q\mathbf{T}(\theta_k)), E_{G^s}(s_q\mathbf{T}(\theta_k))]^\mathsf{T} - \\ &\qquad \sum_{i=1}^{3}(\beta_i^k-\alpha_i^k)E_F(\mathbf{T}(x_{k_i}))|_1\\ &s.t. \quad \forall k,l, \quad \sum_q G^s_{k,l,q} = F_{k,l}, \quad \sum_q G^c_{k,l,q} = F_{k,l}\end{aligned} \quad (11)$$

We define a combined affine cost

$$\text{AffCon}(F,k) = \text{AffCon}_\mathbf{x}(F,k) + \text{AffCon}_\theta(F,k) \quad (12)$$

and include this in the overall objective for the histogram flow optimization.

Up to now we have described attributed optical flow, attribute flow and attribute histogram flow with their respective affine transformation constraints. The relationship between three problems is summarized in Figure 4.

### 3.3. Ground Distance in Histogram Flow

Attribute flow matching could have multiple valid solutions of transformation. For example a circle can be matched to itself under any rotation. We introduce a bias to pick one such transformation using the principle of least action. We borrow the idea from Earth Mover's Distance (EMD) [8], which minimizes the cost of histogram flow transportation subject to constraints on preservation of flow.

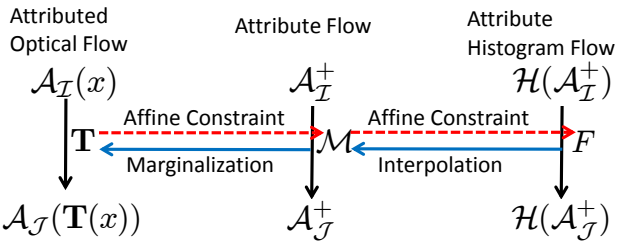


Figure 4. Given affine constraints on spatial transformation $\mathbf{T}$, equivalent constraints on the attribute flow $\mathcal{M}$ is formed and further passed on to the histogram flow $F$. We solve for the optimal histogram flow $F$ under these affine constraints, and then recover the spatial transformation $\mathbf{T}$.

The cost of flow transportation is parametrized by ground distance $d_{k,l} \geq 0$ for each flow $F_{k,l}$. The cost of the flow from histogram $\mathcal{H}(\mathcal{A}^+_\mathcal{I})$ to histogram $\mathcal{H}(\mathcal{A}^+_\mathcal{J})$ is defined as:

$$\text{GD}(F) = \sum_{k,l=1}^{m} F_{k,l}d_{k,l} \quad (13)$$

In our experiments, we use $L_2$ Euclidean distance between histogram bin locations as the the ground distance.

## 4. Matching with Selected Regions

In our problem of object recognition, image $\mathcal{J}$ will have a large image background. It is often possible to cherry pick (selectively choose) the 'good' image attributes in $\mathcal{J}$'s background so that they can be matched to the model. Since we allow image deformation, the risk of picking background attributes is even greater.

Our observation is that on $\mathcal{J}$, the image attributes are correlated through underlying image grouping structure (e.g. salient contours grouped by edges, large segments grouped by pixels). These structures tend to be foreground or background as a whole.

To take advantage of the grouped image components, we actively *select* the image contours to be used in the deformable correspondence. Once a contour is selected, all attributes it carries will be used for matching. This reduces the risk of 'cherry picking' significantly.

Let $\mathcal{C}$ be a set of contours. We want to simultaneous select foreground image contours and solve image deformation via histogram flow computation. We introduce contour selection indicator $\mathbf{x}^{\text{sel}} \in \{0,1\}^{|\mathcal{C}|}$, where $\mathbf{x}^{\text{sel}}_i = 1$ iff contour $C_i$ is selected to be foreground.

With contour selection, the attribute histogram of $\mathcal{J}$ is a function of the selected image contours: $\mathcal{H}(\mathcal{A}^+_\mathcal{J}, \mathbf{x}^{\text{sel}})$. As recognized by [14], this quantity can be defined via per-contour histogram matrix $\mathcal{H}_\mathcal{J} \in \mathbb{R}^{m\times|\mathcal{C}|}$ :

$$\mathcal{H}(\mathcal{A}^+_\mathcal{J}, \mathbf{x}^{\text{sel}}) = \mathcal{H}_\mathcal{J}\mathbf{x}^{\text{sel}} \quad (14)$$

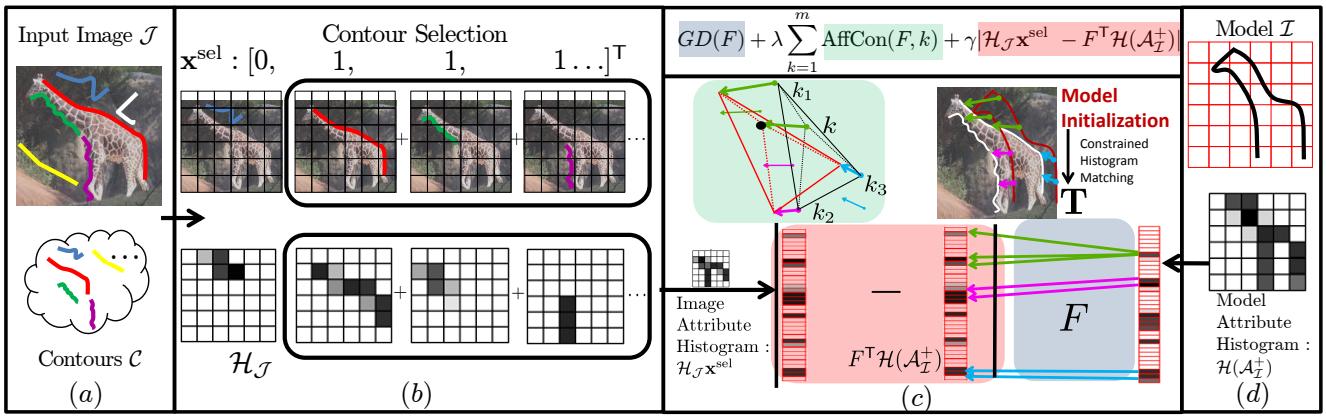


Figure 5. **Illustration of solving constrained attribute histogram flow.** The input are (a):test images and contours and (d): model and histograms. (b) illustrates the contour selection by adding up histogram over selected image contours. (c) illustrates the constrained attribute histogram flow. Each term in cost function is visualized via the region highlighted with the corresponding color. Colored arrows shows the correspondence between optical flow (top right), histogram flow (bottom right) and affine constraint (top left).

The $i$-th column of $\mathcal{H}_{\mathcal{J}}$ corresponds to a histogram over the oriented image edges of image contour $C_i$.

## 5. Constrained Histogram Flow

Incorporating affine constraint costs, ground distance and foreground contour selection into histogram flow in (Eq. 6), as shown in Figure 5, we define the **Constrained Histogram Flow** problem:

$$\begin{aligned} \min_{F,\mathbf{x}^{\text{sel}}} \quad & \text{GD}(F) + \lambda \sum_{k=1}^{m} \text{AffCon}(F,k) \\ & + \gamma |F^{\mathsf{T}}\mathcal{H}(\mathcal{A}_{\mathcal{I}}^{+}) - \mathcal{H}_{\mathcal{J}}\mathbf{x}^{\text{sel}}|_1 \\ \text{s.t.} \quad & F\mathbf{1} = \mathbf{1}, \quad F \geq 0, \quad \mathbf{x}^{\text{sel}} \in [0,1]^{|\mathcal{C}|} \end{aligned} \tag{15}$$

The contour selection indicator vector $\mathbf{x}^{\text{sel}}$ is relaxed to be continuous in $[0,1]^{|\mathcal{C}|}$. Both objective and constraints can be encoded in a single linear programming, which can be efficiently solved with off-the-shelf linear program solvers.

The variable size is dominated by the dimension of flow variables. Although there can be $m^2$ possible flows with $m$ histogram bins, we can prune flow between far away bins, considering only limited translation. Parameters $\gamma, \lambda$ balance the ground distance, affine cost and histogram difference. We use $\gamma = 0.3$, $\lambda = 0.1$ throughout our experiment.

## 6. Object Alignment and Detection

We test our method on the ETHZ Shape Classes Dataset [2] with five categories: Applelogos, Bottles, Giraffes, Mugs and Swans. We first use the method of Srinivasan et al. [10] to generate a shortlist of detection in each image for each object category. We initialize the model location from detection bounding box and solve constrained histogram flow with contour selection.

**Structural Distance Preserving**:

During the detection we only consider a subset of all possible affine transformations. We disallow large scale changes, which we encode using a constraint that limits $L_1$ distance between transformed model points. We also restrict the rotation by at most $\pm\pi/2$, which we encode using a constraint that preserves the sign of horizontal coordinate differences between transformed model points. Both constraints can be expressed linearly via the expected transformation $E_F(\mathbf{T}(x))$.

**Learning in Canonical Model Space:**

After solving constrained histogram flow $F$ to estimate $\mathbf{T}$, we deform the test image to align with the model shape. We train a discriminative classifier to perform a final verification in the canonical model space. Since pose variation have been eliminated with the alignment step, we can learn from fewer training examples. Furthermore, the learning algorithm can pick up smaller but distinctive features which are often hard to pick up due to misalignment between the training examples.

**Implementation Details and Results:**

We implemented our method in MATLAB using the MOSEK linear programming solver. Bottom-up contours are extracted in each image using the method of [13]. During detection, we use the contour model learned in [10] to generate detection candidates and solve the constrained histogram flow to estimate the deformation.

We quantize 2D space into 5x5 pixels grid, and edge orientation into 8 angular bins. We consider a global affine transformation for each model class. We select three anchoring bins with non-zero count that enclose the largest triangle area. Each candidate alignment takes about 2s to solve.

To score each detection, we warp all the contours in-

side the bounding box back to model space, instead of just the selected ones in constrained histogram flow to allow for richer feature for discrimination.

We solve joint selection on warped contours using the learned histogram bin weights to get the final detection score. We use the feature consists of the absolute value of the bin count differences between the model histogram, and the image histogram for a particular selection of warped image contours. More details about joint selection and weight learning are described in [10].

| | Applelogos | Bottles | Giraffes | Mugs | Swans | **Mean** |
|---|---|---|---|---|---|---|
| **Attribute Flow** | 0.930 | 0.977 | 0.783 | 0.895 | 0.972 | 0.911 |
| Ma et al. [6] | 0.881 | 0.920 | 0.756 | 0.868 | 0.959 | 0.877 |
| Srini. et al. [10] | 0.845 | 0.916 | 0.787 | 0.888 | 0.922 | 0.872 |
| Maji et al. [7] | 0.869 | 0.724 | 0.742 | 0.806 | 0.716 | 0.771 |
| Lu et al. [5] | 0.844 | 0.641 | 0.617 | 0.643 | 0.798 | 0.709 |
| Toshev et al.§ [12] | 0.983 | 0.936 | 0.713 | 0.718 | 0.973 | 0.865 |

Table 1. Comparison of **interpolated average precision (AP)** on the ETHZ shape categories dataset. Our method has the highest mean AP across categories. § uses a different train-test split.

We compare out results against the reported ones in [10], [7], [5], [6] with the same train/test split, and with [12] with a different one. For each category, during training we use the first half of the images from that category as positive examples and sample the same number of negative examples from remaining categories equally. The rest images are all for testing. [12] split half of the entire dataset for training, which leaves fewer test images than us. We use 0.5 overlap score threshold during the comparison. Table 1 shows the interpolated average precision. We outperform in mean performance compared the previous state-of-the-art result [10], [6]. We shown the precison/recall (PR) curves compared with [10], [7], [5] in Figure 6 as well as the false positives per images (take log as x axis). Figure 7 shows the segmented result and the model correspondence.

## 7. Related Work

Our work is related to 1) optical flow methods which compute a dense correspondence between two images, and 2) feature-based methods that compute sparse correspondences. In contrast to optical flow approaches we allow highly discriminative features (e.g. edge orientation, histogram of gradient ) which are sensitive to transformation.

Sparse feature correspondence is typically formulated as a graph matching problem [9, 8, 11], with the graph matching cost consisting of unary terms measuring local appearance similarity and pairwise terms measure the geometrical consistency of two different matchings.

Typical pairwise constraints preserve distances and angles between feature points. As such they only allow rigid transformations between the two images. Li et al. [3] described a method for encoding local affine transformation constraints on the graph matching space. Similarly, Jiang and Yu [4] introduced a method for encoding a global similarity transform constraint on the graph matching space.

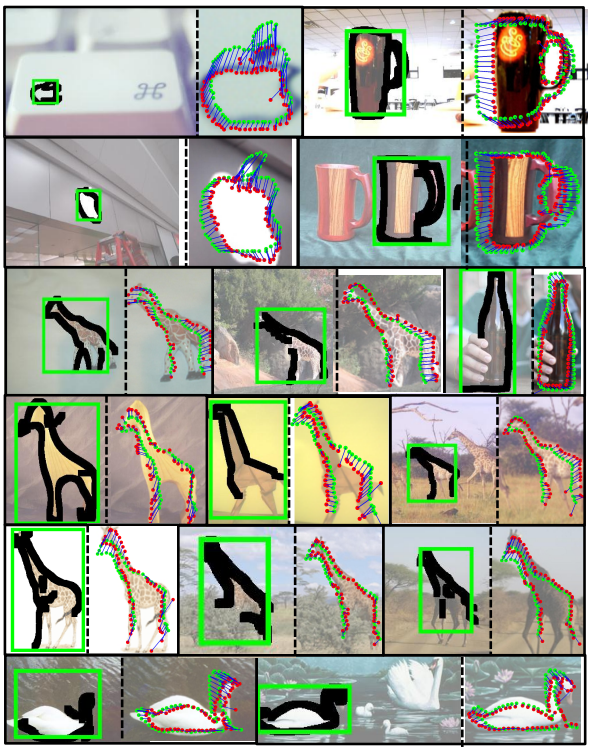

Figure 7. **Some detection and model alignment results on ETHZ shape categories dataset.** Detections and estimated model deformation are shown side by side. On the left we show the selected objects contours (black) and predicted bounding box (green). On the right we show the input rigid object model (green dots) and deformed model(red dots) estimated from attribute flow (blue lines).

While the global constraints on the transformation that explain the local feature matching improve the matching result in many cases (particularly for planar objects that undergo out-of-plane rotation), none of these previous graph matching works address the fact that the features themselves must undergo deformation, changing with different hypotheses for the affine transformation relating the two images. One of the key contributions of our work is encoding exactly this property in our image features (attributes) and our matching between the sets of image features.

Our computational solution with histogram has some similarity to the EMD [8]. The EMD allows many-to-many correspondence of histogram bins, thereby allowing us to compute dense correspondences. However, traditional EMD has no geometric constraints on the flow, and hence no guarantee that the resulting transformation computed from the flow has a valid spatial interpretation.

Another important issue of image matching is avoiding accidental alignments of the object model to background

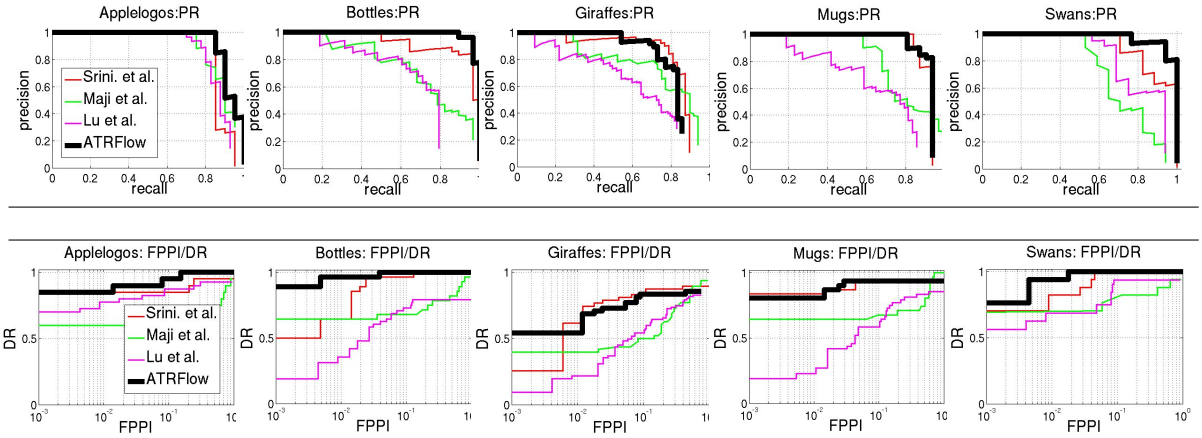


Figure 6. **Precision vs. recall** (PR: top row) and **false positives per image** vs. detection rate (FPPI/DR: bottom row) curves for each method. Our methods is labelled "ATRFlow". We take the log of false positive image to strength the high precision region.

clutter. Our approach builds on the bottom-up, many-to-one matching of Zhu et al. [14], which allows us to select foreground bottom-up structures such as image contours to participate in the matching, while removing background clutter contours that can cause accidental alignments. We show that this can be done in one step with a single, unified cost function, yielding highly accurate object detection that fires rarely in background clutter.

## 8. Conclusion

Deformable object recognition is a challenging problem in vision. Current methods model all possible deformation of the object explicitly, because they lack understanding how discriminative but transformation variant image attributes varies under transformation. By formulating the image deformation as an attributes flow, we are able to explain the attribute variation without explicit image deformation. We are able to achieve this by imposing constraints on the attribute flow to ensure it has a valid geometry interpretation. We approximate the attribute flow using non-parametric histogram flow, which can be solved efficiently using linear programming. We verify our methods on ETHZ shape classes dataset.

## References


[1] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(1):36–51, 2008. 2394

[2] V. Ferrari, F. Jurie, and C. Schmid. Accurate object detection with deformable shape models learnt from images. In *CVPR*, 2007. 2398

[3] H.Li, E. Kim, X. Huang, and L. He. Object matching with a locally affine-invariant constraint. In *CVPR*, 2010. 2399

[4] H. Jiang and S. X. Yu. Linear solution to scale and rotation invariant object matching. In *CVPR*, 2009. 2399

[5] C. Lu, L. J. Latecki, N. Adluru, H. Ling, and X. Yang. Shape guided contour grouping with particle filters. In *ICCV*, 2009. 2399

[6] T. Ma and L. J. Latecki. From partial shape matching through local deformation to robust global shape similarity for object detection. In *CVPR*, 2011. 2399

[7] S. Maji and J. Malik. A max-margin hough tranform for object detection. In *CVPR*, 2009. 2399

[8] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000. 2397, 2399

[9] G. L. Scott and H. C. L. Higgins. An algorithm for associating the features of two images. *Proceedings: Biological Sciences*, 244(1309):pp. 21–26, 1991. 2399

[10] P. Srinivasan, Q. Zhu, and J. Shi. Many-to-one contour matching for describing and discriminating object shape. In *CVPR*, 2010. 2398, 2399

[11] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. In *ECCV*, 2008. 2399

[12] A. Toshev, B. Taskar, and K. Daniilidis. Object detection via boundary structure segmentation. In *CVPR*, 2010. 2399

[13] Q. Zhu, G. Song, and J. Shi. Untangling cycles for contour grouping. In *ICCV*, 2007. 2398

[14] Q. Zhu, L. Wang, Y. Wu, and J. Shi. Contour context selection for object detection: A set-to-set contour matching approach. In *ECCV*, 2008. 2394, 2397, 2400