ProAlignNet : Unsupervised Learning for Progressively Aligning Noisy Contours

VSR Veeravasarapu, Abhishek Goel, Deepak Mittal, Maneesh Singh
Verisk AI, Verisk Analytics
{s.veeravasarapu, a.goel, d.mittal, m.singh}@verisk.com

Abstract

Contour shape alignment is a fundamental but challenging problem in computer vision, especially when the observations are partial, noisy, and largely misaligned. Recent ConvNet-based architectures that were proposed to align image structures tend to fail with contour representation of shapes, mostly due to the use of proximity-insensitive pixel-wise similarity measures as loss functions in their training processes. This work presents a novel ConvNet, "ProAlignNet," that accounts for large scale misalignments and complex transformations between the contour shapes. It infers the warp parameters in a multi-scale fashion with progressively increasing complex transformations over increasing scales. It learns –without supervision– to align contours, agnostic to noise and missing parts, by training with a novel loss function which is derived an upperbound of a proximity-sensitive and local shape-dependent similarity metric that uses classical Morphological Chamfer Distance Transform. We evaluate the reliability of these proposals on a simulated MNIST noisy contours dataset via some basic sanity check experiments. Next, we demonstrate the effectiveness of the proposed models in two real-world applications of (i) aligning geo-parcel data to aerial image maps and (ii) refining coarsely annotated segmentation labels. In both applications, the proposed models consistently perform superior to state-of-the-art methods.

1. Introduction

Contour shape alignment with noisy image observations is a fundamental, but challenging, problem in computer vision and graphics fields, with diverse applications including skeleton/silhouette alignment [2] (for animation retargeting), semantic boundary alignment [28] and shape-to-scan alignment [15] etc. For instance, consider the first row of Figure 1. It represents a process of geo-parcel alignment that requires aligning geo-parcel data (legal land boundaries maintained by local counties) to aerial image maps. These two modalities of geo-spatial data, if well aligned, are useful to assist the processes of property assessment and tax/insurance underwritings. Classically, contour alignment

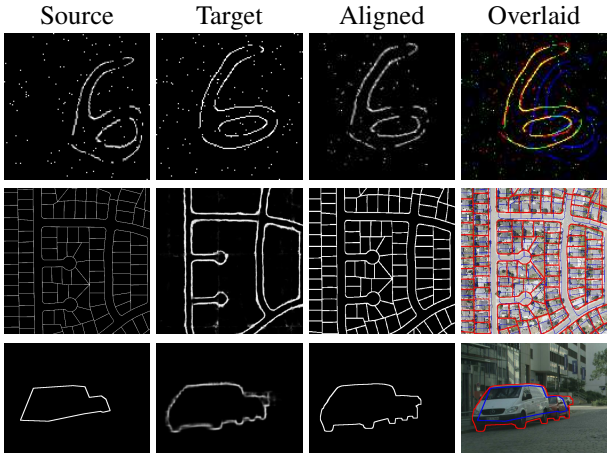


Figure 1: This work considers the problem of learning to align source (1st column) with target (2nd column) contour images. Aligned results are shown in 3rd column. Visualizations (4th column) are input image canvases overlaid with original (Blue) and aligned (Red) source contours. Three rows contain sample results from the applications we considered in this work: (i) noisy digit contour alignment (ii) geo-parcel alignment and (ii) coarse-label refinement.

problems have been approached by finding key points or features of the shapes and aligning them by optimizing for the parameters of a predefined class of transformations such as affine or rigid transforms [25]. These methods may not work for the shapes whose alignment requires a transform different from the hired ones. Nonrigid registration methods have also been proposed in the literature, mainly using intensity-based similarity metrics [19]. However, these methods are often computationally expensive and sensitive towards corrupted parts of the shapes.

Motivated by the strong invasion and great success of deep convolutional neural networks (ConvNets) in various vision tasks, some recent works [17, 20, 13, 14, 7] have designed ConvNet based architectures for shape alignment/registration and shown impressive results on the datasets with limited misalignments. However, we observe that these approaches tend to fail in noisy, partially observed and largely misaligned contour shape contexts. We

believe that this is due to (a) direct prediction of alignment warp in one-go at input resolution and (b) training with proximity-insensitive pixel-level similarity metrics (such as normalized cross correlation [7]) that might result in non-informative gradients as the spatial overlap between contours may not be significant.

To overcome the problems mentioned above, we propose a novel ConvNet architecture, "***ProAlignNet***" that learns to align a pair of contour images while being robust to noise and partial occlusions in the images. It finds an optimal transformation that aligns source to target contours in a multi-scale fashion with progressively increasing complex transformations over increasing scales.

We also propose a novel loss function that accounts for not only local shape similarity but also proximity of similar shapes. The proposed loss is based on classical *Chamfer* distance that measures the proximity between the contours, thus, provides informative gradients even though there is no spatial overlap between the contours. Chamfer distance was a popular metric [3, 11] for binary shape alignment and was traditionally implemented efficiently using morphological distance transforms [5]. However, this morphological chamfer distance transform (MCDT) is nondifferentiable wrt the warp parameters, which makes it nontrivial to use with BackProp. Hence, we device a reparameterization trick (inspired from homeomorphism properties of the transformation families we use) to make it usable with BackProp. However, this trick requires a backward/inverse transform (that aligns target to source) to be measured. Here, we propose to swap the roles of source and target features in the alignment modules to get the backward transform with the same network components without any additional parameters/layers. As a side benefit, this forward-backward transform consistency constraint acts as a powerful regularizer that improves generalization capabilities.

Chamfer distance is amenable to noise as it uses Euclidean distance to find nearest neighbours, and it neglects local shape statistics around pixels. Hence, we introduce a data-dependent term into distance function that MCDT uses. This modification increases the proposed loss's robustness to noise pixels. We then derive an upperbound of this loss function, which is differentiable and computationally efficient. Since we use one instance of the loss at each scale, this multiscale objective passes gradients to all the layers simultaneously. It helps the training process to be more stable and less sensitive to hyperparameters such as learning rates as reported in [18]. Our training process is completely unsupervised as it does not require any ground-truth (GT) warps that the network is expected to reproduce. By training with a loss function that accounts for local shape contexts while being invariant to corrupted pixels, our network learns to be robust to noise and partial occlusions.

In summary, our contributions are the following:

- **ProAlignNet**: a novel multiscale contour alignment network that employs progressively increasing complex transforms over increasing scales.
- **Shape-sensitive Chamfer Upperbound Loss**: a novel loss function that measures proximity and local shape similarity while being robust towards noise and partial occlusions.
- Ablation studies with **MNIST noisy digit contours**.
- Demonstration of efficacy of the proposed models in two real-world applications: **geo-parcel alignment** and **coarse-to-fine label refinement**.

2. Motivations and Background

Deep Learning for Shape Alignment: Several works have employed deep networks [17, 20, 13] to directly estimate warp parameters and trained using ground-truth warp fields/parameters. However, it is challenging to collect ground-truth warp fields for several real-world applications, especially for nonrigid alignment scenarios. Hence, recent works [14, 7] propose unsupervised methods for nonrigid registration and these are closest to our approach. The work of [7] designed a deep network referred to as DIRNet, consists of a CNN based regressor that predicts a deformation field, followed a spatial transformation function that warps source to target image. The regressor takes concatenated pairs of source and target images and predicts displacement vector fields directly from convolutional features. These fields are then upsampled using splines to original resolutions and used to warp the source images. These models were trained with pixel-wise similarity metrics; thus, they can deal only with small scale deformations. Similarly the work of [14] proposed a ConvNet referred as AlignNet which constitutes a regression network and a novel integrator layer that outputs free-form deformation field. We empirically show that these approaches might fail in the contexts of contour shapes with large misalignments. Unlike these methods, our ProAlignNet uses inference process that accounts for large scale misalignments and complex transformations between the contour shapes by inferring in a multi-scale fashion with progressively increasing complex transformations over increasing scales.

Chamfer distance: Proximity metrics such as Chamfer distances and Earth-mover distances [12] are more popular in shape correspondences. They are criticized for their computationally cost. However, when shapes are represented in binary contour images, morphological distance transforms [22] can be used to compute the Chamfer distance between two contour images efficiently. However, it has two problems in its vanilla form: (i) nondifferentiability: morphological distance transform has a process of collecting all nonzero pixels to find the nearest one. This set-collection operation is nondifferentiable; (ii) sensitivity towards noise:

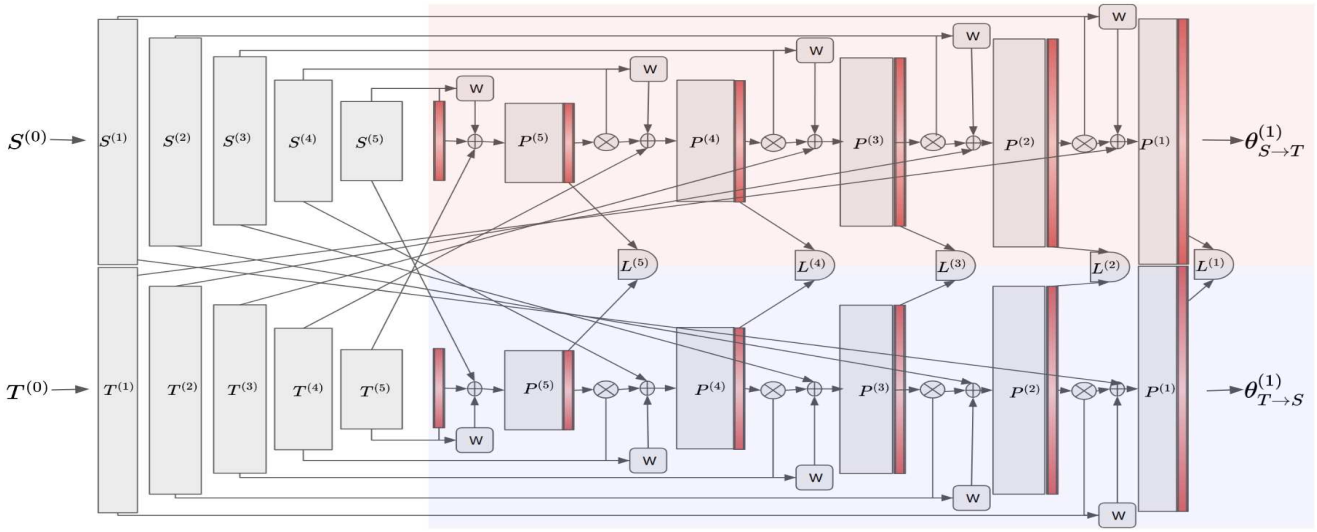


Figure 2: Overview of ProAlignNet with 5 Scales: Given a pair of source ($S^{(0)}$) and target ($T^{(0)}$) contour images, we first use base CNNs to get the feature maps. A cascade of Warp predictors ($P^{(i)}$) operates in a multiscale fashion to predict transformations to align source to target features. Coarsest predictor starts with identity transformation and uses affine transform to refine it. Complexity of transformation used in these predictors progressively increases over scales. Finest scale predictor used more complex transformation such as thin-plate-splines with higher number of control points.

while computing distance to the nearest pixel, it is blind to noise pixels. This influences the distance estimate between shapes. In this work, we derive a loss function, which is based on Chamfer distance but overcomes problems mentioned above.

3. ProAlignNet

The network architecture of the *ProAlignNet*, as shown in Figure 2, consists of a set of simple modules: (i) Base CNN blocks to extract features ($S^{(i)}$ and $T^{(i)}$) of source and target images at different scales; (ii) Warp predictors ($P^{(i)}$) that predict warp fields with a predefined class of transformations at each scale; (iii) Warp layer (W) that warps the features using warp fields. Here, i represents the index of scale with $i = 0$ and $i = K$ denoting the original (finest) and coarsest scales respectively.

3.1. Base CNNs for Feature Extraction

Given the source and target input image pair $(S^{(0)}, T^{(0)}) \in R^{h\times w\times c}$, we extract their feature representations $\left\{\left(S^{(i)}, T^{(i)}\right)_{i=1}^{K}\right\}$ from different layers $i \in \{1, 2, ..., K\}$ of fully-convolutional backbone networks F_s and F_t. As shown in Figure 2, we use $K = 5$ in all experiments provided in this work.

3.2. Warp Predictors and Warp Layer

Once we extract the features from both source and target, we design a cascade of warp predictors ($P^{(i)}$) that learn to predict transformations to align source to target features at multiple scales. Each $P^{(i)}$ block takes a feature tensor that is a result of concatenating (denoted by $\oplus$ in Figure 2) three features: (i) source features warped by the warp field from previous coarser scale but upsampled (denoted by $\otimes$) by a factor of 2, (ii) target features, and (iii) upsampled warp field from previous coarser scale. Mathematically, warp predictor at a given scale i learns the below functionality,

$$P^{(i)} : S^{(i)}(\theta^{i+1}_{\otimes 2}) \oplus T^{(i)} \oplus \theta^{i+1}_{\otimes 2} \longrightarrow \theta^{(i)}$$

where $\theta_{\otimes u}$ denotes warp field θ upsampled by a factor of u. We initialize θ^{K+1} as identity warp field at coarsest scale K.

A warp or transformer layer (W) uses the predicted warp parameters to derive displacement field and warp the source features. These warp predictors by design resemble Spatial Transformer Networks proposed in [16]. In general, they have a regression network that predicts parameters of the transformations, followed by a grid layer that converts parameters to pixel-level warp field.

In Figure 2, the red shaded part of ProAlignNet estimates a forward transformation that aligns source to target. It initializes the transformation with identity warp grid and refines it using multiscale warp predictors. It starts the refinement process by using simple affine transformation at the coarsest scale, which is a linear transformation with 6 DOF. The estimated affine transform grid is used to align source features using warp layer. The aligned features are then passed along with estimated affine grid through the next finer scale network, which estimates a warp field using more flexible transformations such as thin-plate splines(*tps*). As we move toward the finest scale, we use *tps* transforms with

increasing resolutions of control point grid. The final estimate of the geometric transformation is then obtained from the finest scale predictor that learns to compose all coarser transformations along with local refinements that align its input features. As we explain in the next section, our loss function requires backward transform also to be measured that aligns target to source. This backward transform is estimated using the same warp predictor components but with a second pass (blue shaded part in Figure 2) by reversing the roles of source and target features.

3.3. Multi Scale Loss Objective

We use one instance of the loss function, $L^{(i)}$, at each scale i as shown in Figure 2. Hence, the multiscale objective we use to train ProAlignNet is given as,

$$L_{MS} = \sum_{i=1}^{K} \lambda_i L^{(i)} = \sum_{i=1}^{K} \lambda_i L\left(S^{(0)}\left(\theta^{(i)}_{\otimes 2^i}\right), T^{(0)}\right) \quad (1)$$

$\theta^{(i)}_{\otimes 2^i}$ denotes the warp field predicted at scale i but upsampled by a factor 2^i that brings up the field to be at similar resolution as the input source and target images $S^{(0)}$ and $T^{(0)}$. λ_i is the weighting factor for the loss L from scale i. L can be any alignment loss function, for instance NCC or MSE. However, motivated by the fact that pixel-wise similarity metrics suffer in large scale misalignment by not considering the proximity of shapes we propose a novel loss function in this work.

4. Shape-dependent Chamfer Upperbound

4.1. Chamfer Loss

Chamfer distance was a popular metric to measure the proximity between two shapes or contours. It was initially proposed for point sets. The chamfer distance b/w any two point sets X and Y is given as,

$$C(X,Y) = \frac{1}{N_X}\sum_{x\in X}\min_{y\in Y} E(x,y) + \frac{1}{N_Y}\sum_{y\in Y}\min_{x\in X} E(y,x) \quad (2)$$

where $E(x,y)$ is Euclidean distance between the points x and y. N_X and N_Y denotes the cardinality of the sets X and Y respectively. In the binary image representation of shapes, one can use the concepts of morphological distance transform to efficiently compute the Chamfer distance between two images. Morphological Distance Transform (MDT) computes Euclidean distance to nearest nonzero neighbor for each pixel x in a given contour image I. Thus, it is represented by $dt[I](x) = \min_{i\in I} E(x,i)$. Using MDT, Chamfer distance between the source (S) and target (T) images can be written[1] as $C(S,T) = \frac{1}{N_S} dt[S].T + \frac{1}{N_T} S.dt[T]$. Here, N_S and N_T denotes number of nonzero pixels in S and T respectively. There are several efficient implementations available to compute $dt[.]$ (in Opencv, Scikit-image etc.). Now the chamfer loss b/w warped source and target at any scale (we drop scale i for simplicity) becomes,

$$C\big(S(\theta),T\big) = \frac{1}{N_S} dt[S(\theta)].T + \frac{1}{N_T} S(\theta).dt[T] \quad (3)$$

The gradient of the above loss function requires the distance transform operation dt to be differentiable wrt warped source ($S(\theta)$). Unfortunately, dt is not differentiable as it has a set-collection process to collect all nonzero pixels.

4.2. Reparametrized Chamfer Loss

However, we overcome this problem by using a reparameterization trick inspired by homeomorphism [21] properties of affine/*tps* transformations. This property states that if a forward transformation $\theta_{S\to T} \in \Theta$ aligns S with T then there exists a $\theta_{T\to S}$ also $\in \Theta$ that aligns T with S, given that Θ is a homeomorphic transformation group. This results in a corollary that $dt[S(\theta_{S\to T})].T = dt[S].T(\theta_{T\to S})$ (Please refer to Supplementary for the proofs). With this reformulation, Eq 3 becomes

$$C_r\big(S(\theta),T\big) = dt[S].T(\theta_{T\to S}) + S(\theta_{S\to T}).dt[T] \quad (4)$$

The gradient of the above loss function doesn't require dt to be differentiable. Distance transform (dt) maps can be computed externally and supplied as reference signals. We refer to this loss as Reparametrized Chamfer loss. However, the above loss requires backward transform $\theta_{T\to S}$ to be estimated.

Bi-directional Transform Consistency: For affine, one can analytically compute backward transform ($\theta_{T\to S}$) using matrix inverse. However, it is not trivial to get analytical inverse for *tps* and other fully flexible transforms. Hence, we propose a simple and effective way to get backward transforms by a second pass through warp predictors by swapping the roles of source and target features. The loss in Eq 4 constraints these forward and backward transforms to be consistent with each other. This constraint acts as a powerful regularizer on the network training. In Section 5, we empirically demonstrate that it improves the generalization capabilities of the models.

4.3. Shape-dependent Chamfer Upperbound

Local-shape dependency: The above Chamfer loss (Eq 4) is susceptible to noise and occlusions in the contour images as it computes the distance from nearest nonzero neighbor without checking if it is a noise pixel or indeed a part of contour. To make it robust, we incorporate local shape dependency into the distance computation to make

[1] Here dot (.) represents a scalar product.

distance transform (dt) to choose nearest pixels based on not only spatial proximity but also local shape similarity. Although several sophisticated local shape metrics are available, we stick to first-order intensity gradients [26, 4] in this work for computational simplicity. More specifically, we consider unit gradients as a representation of local orientations of the contour pixels. We leave advanced local shape metrics for future work. Here, we use a combination of Euclidean distances in Cartesian and image gradient space. Now local shape-dependent Chamfer distance is given by,

$$C_d(X,Y) = \frac{1}{N_X}\sum_{x\in X}\min_{y\in Y}\Big(E(x,y)+\alpha E(I'_x,I'_y)\Big) + \frac{1}{N_Y}\sum_{y\in Y}\min_{x\in X}\Big(E(y,x)+\alpha E(I'_y,I'_x)\Big) \tag{5}$$

where I'_x denotes the unit gradient vector computed at x.

Upperbound of shape-dependent Chamfer loss: However, we can not use the concept of MDT here as in Eq 4 as the distance has a data-dependent term. Fortunately, min arguments before distance terms in Eq 5 allow us to use a famous math property known as "*min-max inequality*." It results in an upperbound for Eq 5 with two simple terms.

We use min-max inequality to reformulate Eq 5 so that one can use the concepts of MDT and reparameterization as in Eq 4. Min-max inequality states that minimum of the sum of any two arbitrary functions $f(x)$ and $g(x)$ is upperbounded by the sum of minimum and maximum of individual functions, i.e., $\min_x\big(f(x)+g(x)\big) \leq \min_x f(x) + \max_x g(x)$ (Please refer to Supplementary for the proofs).

Using the inequality for both terms on RHS of Eq 5 results an upperbound with original Chamfer distance (Eq 3) and shape-dependent term as follows,

$$C_d(X,Y) \leq C(X,Y) + \alpha\Big(\frac{1}{N_X}\sum_{x\in X}\max_{y\in Y}E(I'_x,I'_y)+\frac{1}{N_Y}\sum_{y\in Y}\max_{x\in X}E(I'_y,I'_x)\Big) \tag{6}$$

Rewriting the above upperbound in the current context of warped source to target alignment,

$$C_d(S(\theta),T) \leq C(S(\theta),T) + \alpha\Big(\frac{1}{N_{S(\theta)}}\sum_{x\in S(\theta)}\max_{y\in T}E(I'_x,I'_y)+\frac{1}{N_T}\sum_{y\in T}\max_{x\in S(\theta)}E(I'_y,I'_x)\Big) \tag{7}$$

We denote this upperbound as C_{up}. As one can observe, the shape-dependent terms are computationally heavy as the maximum being taken over the window of the entire image for each pixel in the other image. However, we can constraint this window to be local and search in the neighborhood defined by that window. Moreover, this maximum-finding operation can be implemented with *MaxPool* layers.

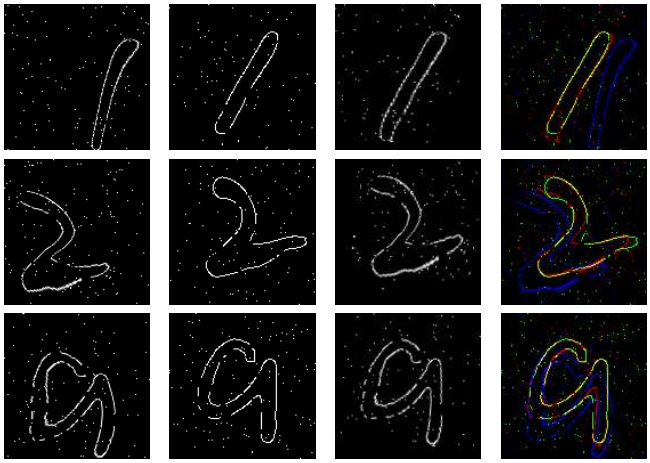
Finally, this local shape-dependent Chamfer upperbound is given by,

$$C_{up}(S,T) = \Big(\frac{1}{N_S}dt[S].T(\theta_{T\to S})+\frac{1}{N_T}S(\theta_{S\to T}).dt[T]\Big) + \alpha\Big(\frac{1}{N_{S(\theta)}}\sum_{x\in S(\theta)}\max_{y\in T_x}E(I'_x,I'_y)+\frac{1}{N_T}\sum_{y\in T}\max_{x\in S_y(\theta)}E(I'_y,I'_x)\Big) \tag{8}$$

When the local window is restricted to be 1×1, minimizing the above term can be related to maximizing cross-correlation in intensity gradient space. When raw pixel intensities are used in place of gradients, this is maximizing NCC-related metric. Now the upperbound loss with unit gradients as local shape measures,

$$C_{up}\big(S(\theta),T\big) = \Big(\frac{1}{N_S}dt[S].T(\theta_{T\to S})+\frac{1}{N_T}S(\theta_{S\to T}).dt[T]\Big) + \alpha\Big(\frac{1}{N_{S(\theta)}}\sum_{x\in S(\theta)}\max_{y\in T_x}\sqrt{1-I'^T_x.I'_y} + \frac{1}{N_T}\sum_{y\in T}\max_{x\in S_x(\theta)}\sqrt{1-I'^T_y.I'_x}\Big) \tag{9}$$

Please refer to Supplementary for detailed derivations of the above equations.



(a) Source (b) Target (c) Aligned (d) Overlaid

Figure 3: Sample results on contourMNIST: First, second and third column images are source, target and aligned source contour images respectively. Right most are the images prepared for better visualization with B,G,R channels as source, target, aligned images respectively.

5. Ablation Studies using contourMNIST

We first evaluate the behavior of our proposals on a simulated dataset. Several recent works [10, 9] adopted MNIST

Application	Method	Loss	Error ($\%px \leq Z$)	Metric
contourMNIST	Given test pairs	-	10.20 (39%)	Chamfer Score: lower the better.
	DIRNet [7]	Normalized Cross-Correlation	8.10 (46%)	
	DIRNet [7]	Local Shape-dependent Chamfer Upperbound	4.83 (69%)	
	ALIGNet [14]	Mean Squared Error	6.69 (55%)	
	ALIGNet [14]	Local Shape-dependent Chamfer Upperbound	4.04 (71%)	
	ProAlignNet	Normalized Cross-Correlation	5.46 (64%)	
	ProAlignNet	Asymmetric Chamfer	3.38 (89%)	
	ProAlignNet	Reparametrized bi-directional Chamfer	2.91 (93%)	
	ProAlignNet	Local Shape-dependent Chamfer Upperbound	**2.17 (96%)**	
Geo-parcel alignment	Given test pairs	-	42.38 (48%)	Chamfer score: lower the better.
	DIRNet [7]	Normalized Cross-Correlation	41.36 (48%)	
	DIRNet [7]	Local Shape-dependent Chamfer Upperbound	27.70 (65%)	
	ALIGNet [14]	Mean Squared Error	39.98 (59%)	
	ALIGNet [14]	Local Shape-dependent Chamfer Upperbound	31.37 (62%)	
	ProAlignNet	Normalized Cross-Correlation	32.78 (61%)	
	ProAlignNet	Asymmetric Chamfer	28.05 (67%)	
	ProAlignNet	Local Shape-dependent Chamfer Upperbound	**20.63 (75%)**	

Table 1: Quantitative evaluations: Chamfer scores are reported along with percentage (in the brackets) of pixels whose misalignment is less than Z pixels. We use $Z = 5$ for MNIST shape alignment and $Z = 20$ for geo-parcel alignment.

dataset [8] to simulate toy datasets to understand the behavior of the models and training processes without requiring lengthy training periods. Following this trend we also simulate a dataset adopting MNIST digits to the current context of noisy contour alignment.

Simulating contourMNIST dataset: MNIST database contains 70000 gray-scale images of handwritten digits of resolution 28×28. The test images (10,000 digits) were kept separate from the training images (60,000 digits). In our experiments, each digit image is upsampled to 128×128 resolution and converted as a contour image. Each contour image is then transformed with randomly generated *tps* transformations to create a misaligned source contour image while the original one considered as the target image. We add noise and occlusions randomly to simulate partial noisy observations of the contours, as shown in Figure 3. Now the task considered in this section is to align these noisy source to target shapes.

Implementation details: To implement the ProAlignNet architecture for this task, we use CNN bases with five convolutional blocks. Each block is composed with three conv (with filter size 5×5) + leakyReLU (with slope 0.1) layers followed by a MaxPool (with stride 2 and kernel size 2×2) layer. Multiscale alignment part consists of five warp predictors ($P^{(i)}$) that operate on five scales and predict increasingly complex transforms. In this application, we use affine transforms at the coarsest scale and *tps* with control grids of increasing resolutions (2×2, 4×4, 8×8, 16×16) at finer scales. The warp predictor blocks are composed of three conv layers and a fully connected MLP. Each conv layer is followed by leakyRelu + MaxPool except the last one. MLP takes in the flatten conv features from the last layer and predict parameters required for the transform at that scale. For instance, *tps* warp predictor outputs $2n + 6$ parameters while affine predictor output 6 parameter values. We set $\lambda_i = 1.0$ and $\alpha = 1e - 2$ for all the experiments in this paper.

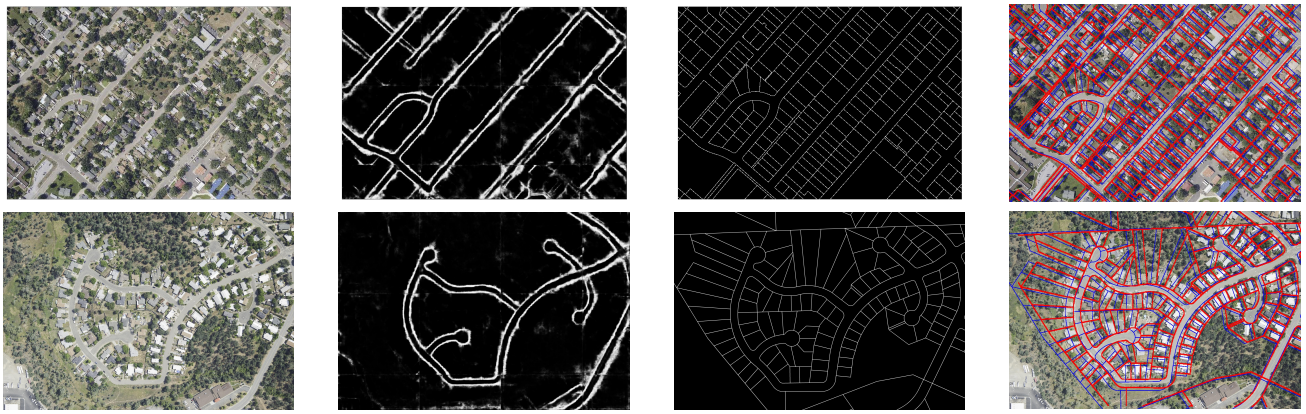
Training: We train the networks for 10 epochs in a completely unsupervised manner using the losses mentioned in Section 4. We use SGD optimizer with learning rate as $1e - 5$.

Evaluation metric: We use asymmetric Chamfer distance b/w aligned source and target images (non-noisy versions), $\frac{1}{N} S(\theta_{S \to T}).dt[T]$, as it measures average misalignment in the units of pixels. We also measure the percentage of pixels whose misalignment under 5 pixels. As shown in Table 1, test set image pairs are misaligned by an average of 10.20 pixels. The percentage of pixels with misalignment under 5 pixels is 39%.

We now evaluate the different aspects of our proposals via a set of experiments.

ProAlignNet vs Baselines : Here we start by comparing the performance of ProAlignNet with that of two baselines, DIRNet [7] and ALIGNet [14], when trained with different loss functions, including the ones used in their implementations and our Chamfer loss functions. Results in Table 1 show that ProAlignNet outperforms DIRNet and ALIGNet in all experimental settings. The best performance is achieved with ProAlignNet model trained using Chamfer uppperbound loss. This is atleast 25% better than the models of DIRNet or ALIGNet. We believe that this superior performance is due to multiscale inference and progressive alignment processes in our network architecture.

Chamfer upperbound vs Other loss functions: The test performances of both DIRNet and ALIGNet has been boosted up by 23% and 16% respectively when trained with



(a) Aerial image (b) Road contours from (a) (c) Geo-parcels (d) Aligned parcels overlaid on (a)

Figure 4: Geo-parcel alignment with road evidences. Better see in enlarged version.

chamfer upperbound loss rather than the ones (NCC, MSE) used in their implementations.

Bidirectional vs Asymmetric Chamfer loss: ProAlignNet yields 4% better performance when trained with bi-directional reparametrized Chamfer loss compared to asymmetric Chamfer loss. This improvement demonstrates the regularization capabilities of forward-backward consistency constraints in the loss. Overall, the best performance of 96% is achieved when ProAlignNet is trained with shape-dependent Chamfer upperbound. This is approximately 7% better than the model that is trained with asymmetric loss.

Impact of local shape dependency: Similarly, the model trained by shape-dependent Chamfer loss performs better than the one trained with bi-directional Chamfer loss by approximately 3%.

6. Geo-parcel to Aerial Image Alignment

Problem statement: In this section, we discuss aligning geo-parcel data to aerial images. Geo-parcel data is used to identify public and private land property boundaries. Parcels are shapefiles with lat-long gps coordinates of the property boundaries maintained by local counties. One can project these parcel shapes (using perspective projection) onto the coordinate system of the camera with which aerial imagery was captured. This process results in binary contour images as shown in Figure 4c. These contours are ideally expected to match visual contours in the aerial image of the corresponding region (shown in Figure 4a). However, due to several differences in their collection processes, these two modalities of the data often misalign by a large extent, sometimes, in the order of 10 meters. Figure 4d depicts the misalignment of the original (before alignment) parcel contours overlaid on the aerial image in blue color. In this work, we extract the road (including sidewalks) contours from aerial images using an off-the-shelf contour detection method [27] and consider these as target contours. These extracted contours are noisy and partial in shape, as seen in Figure 4b. We train our ProAlignNet to align parcel contours with these road contours from aerial imagery.

Dataset: Our dataset contains 1189 aerial and parcel image pairs captured over residential areas of Redding, California. Fortunately, our method does not require any ground-truth alignment parameters to train. However, we prepare a validation set for which we manually aligned 27 parcel-aerial image pairs with more than 7000 parcel polygons.

Implementation details: Experimental settings are similar to the above section. However, we work with input resolution of 1024×512 resolution for this application.

Evaluation metric: In addition to average misalignment (Chamfer score), we also report the percentage of pixels with misalignment under 3ft (20 pixels as ground-sampling-distance is 4.5cm/px for our aerial data).

Evaluation: Parcel data aligned with ProAlignNet (trained with Local Shape-dependent Chamfer Upperbound) is overlaid with red color in Figure 4d. As one can see, it is aligned well with the aerial image contents than the original parcels (blue color). Results in Table 1 show that ProAlignNet outperforms both DIRNet and ALIGNet even when trained with NCC. Best performance (75%) is achieved when ProAlignNet is trained with Local Shape-dependent Chamfer Upperbound. Alignment quality of these parcels with the corresponding aerial image contents has been improved by 27% with ProAlignNet. Moreover, training with our Chamfer upperbound loss has boosted up the performances of DIRNet and ALIGNet, similar to the behavior observed on contourMNIST data.

7. Refining Coarser Segmentation Annotations

In this section we demonstrate that how the proposed models can be used to refine coarsely annotated segmentation labels. For these experiments we use CityScapes

dataset [6], a publicly available benchmark for traffic scene semantic segmentation. It provides the data in 3 sets with public access to GT labels: *train* (2975 samples), *val* (500 samples), and *train-extra* (19998 samples) sets. The image samples in the larger set, *train-extra*, have only coarser annotations (See Figure 5 & 1), while the sets of *train* and *val* provide both finely and coarsely annotated GT labels. Refining the coarser annotations using ProAlignNet is our task of interest in this section. We use an off-the-shelf semantic contour extraction method, CASENet [27] (pretrained on *train* set) to get contours from the images which are treated as target shapes in the current context. The given coarsely annotated labels are considered as source contours and to be aligned with contour predictions from the CASENet model. Here, we train our ProAlignNet using the chamfer upper-bound loss on *train* set. Experimental set up is similar to above sections except that tps control grid resolutions in $P^{(3)}$ and $P^{(4)}$ (see Figure 2) have been doubled for this application. For quantitative analysis, we use *val* set for which both coarse and fine labels are available. Qualitative results are shown on *train-extra* images.

Figure 5: Refining Coarser Labels: Coarser annotations (blue parts in left image) are refined to extract precise labels (red parts in right image).

Label Quality	4px error	8px error	16px error	32px error	Real Coarse
Num Clicks per Image	175.23	95.63	49.21	27.00	98.78
Test IoU	74.85	53.32	33.71	19.44	48.67
GrabCut [24]	26.00	28.51	29.35	25.99	32.11
STEALNet [1]	78.93	69.21	58.96	50.35	67.43
ProAlignNet (Ours)	79.41	69.73	67.51	61.05	71.45

Table 2: Model trained on *train* set and used to refine coarse data on *val* set. Real Coarse corresponds to coarsely human annotated val set, while x-px error correspond to simulated coarse data. Score (%) represents mean IoU.

Coarse Label Simulation: For training data augmentation and quantitative study (shown in Table 2), we also synthetically coarsen the given finer labels following the procedure described in [1, 29]. This synthetic coarsening process first erodes the finer segmentation mask and then simplifies mask boundaries using Douglas-Peucker polygon approximation method to produce masks with controlled quality. Intersection-over-Union (IoU) metrics b/w these coarser and finer labels of *val* set are shown in Table 2. We also count the number of vertices in the simplified polygon and report it in Table 2 as an estimate of the number of clicks required to annotate such object labels.

Results: A recent work, STEALNet [1], addressed this problem of refining coarser annotation labels. Hence, we use it as a baseline for comparison along with GrabCut tool [24]. As reported in Table 2, our method perform equally well with STEALNet at lower-scale misalignments. However, the superiority of our ProAlignNet is quite evident with larger misalignments (16px, 32px errors). Also, our method is better by $\sim 4\%$ compared to STEALNet on real coarser labels of *val* set,. As shown in Figure 5, by starting from a very coarse segmentation mask our method is able to obtain very precise refined masks. Hence, we think that our approach can be introduced in current annotation tools saving considerable amount of annotation time.

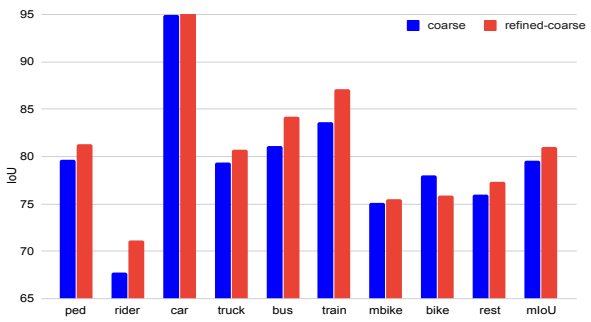


Figure 6: Semantic Segmentation on Cityscapes *val* set: Performance of UNet when trained with (in addition to *train* set) coarse labels vs our refined labels of *train-extra* set. We see improvement of more than 3 IoU % in rider, bus and train.

Improved Segmentation: We also evaluate whether our refined label data is truly beneficial for training segmentation methods. Towards this end, we refine 8 object classes in the whole *train-extra* set. We then train our implementation of UNet based semantic segmentation architecture [23] with the same set of hyper-parameters with and without refinement on the coarse labels of *train-extra* set. Individual performances (IoU%) on the 8 classes are reported in Figure 6. Training with refined labels results in improvements of more than 3 IoU% for rider, bus and train as well as 1.5 IoU% in the overall mean IoU (79.52 vs 81.01).

8. Conclusions

This work introduced a novel ConvNet-based architecture, "*ProAlignNet*," that learns –without supervision– to align noisy contours in multiscale fashion by employing progressively increasing complex transformations over increasing finer scales. We also proposed a novel proximity-measuring and local shape-dependent Chamfer distance based loss. The sanity checks for behaviors of the proposed networks and loss functions have been done using a simulated contourMNIST dataset. We also demonstrated the efficacy of the proposals in two real-world applications: (a) aligning geo-parcel data with aerial imagery, (b) refining coarsely annotated segmentation labels. In future, we explore more effective feature fusing schemes for warp predictors and sophisticated/learnable shape metrics.

Acknowledgments: We thank GEOMNI (www.geomni.com) for providing the data for geo-parcel alignment application.

References

[1] David Acuna, Amlan Kar, and Sanja Fidler. Devil is in the edges: Learning semantic boundaries from noisy annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 8

[2] Dimitrios S Alexiadis, Philip Kelly, Petros Daras, Noel E O'Connor, Tamy Boubekeur, and Maher Ben Moussa. Evaluating a dancer's performance using kinect-based skeleton tracking. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 659–662. ACM, 2011. 1

[3] Gunilla Borgefors. Distance transformations in arbitrary dimensions. *Computer vision, graphics, and image processing*, 27(3):321–345, 1984. 2

[4] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *European conference on computer vision*, pages 25–36. Springer, 2004. 5

[5] M Akmal Butt and Petros Maragos. Optimum design of chamfer distance transforms. *IEEE Transactions on Image Processing*, 7(10):1477–1484, 1998. 2

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Scharwächter, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset. In *CVPR Workshop on the Future of Datasets in Vision*, volume 2, 2015. 8

[7] Bob D de Vos, Floris F Berendsen, Max A Viergever, Marius Staring, and Ivana Išgum. End-to-end unsupervised deformable image registration with a convolutional neural network. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 204–212. Springer, 2017. 1, 2, 6

[8] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 6

[9] Zhiwei Deng, Jiacheng Chen, Yifang Fu, and Greg Mori. Probabilistic neural programmed networks for scene generation. In *Advances in Neural Information Processing Systems*, pages 4028–4038, 2018. 5

[10] SM Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Geoffrey E Hinton, et al. Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems*, pages 3225–3233, 2016. 5

[11] Dariu M Gavrila et al. Multi-feature hierarchical template matching using distance transforms. In *icpr*, 1998. 2

[12] Kristen Grauman and Trevor Darrell. Fast contour matching using approximate earth mover's distance. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. IEEE, 2004. 2

[13] Shaoya Guan, Cai Meng, Yi Xie, Qi Wang, Kai Sun, and Tianmiao Wang. Deformable cardiovascular image registration via multi-channel convolutional neural network. *IEEE Access*, 7:17524–17534, 2019. 1, 2

[14] Rana Hanocka, Noa Fish, Zhenhua Wang, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Alignet: Partial-shape agnostic alignment via unsupervised learning. *ACM Transactions on Graphics (TOG)*, 38(1):1, 2018. 1, 2, 6

[15] David A Hirshberg, Matthew Loper, Eric Rachlin, and Michael J Black. Coregistration: Simultaneous alignment and modeling of articulated 3d shape. In *European Conference on Computer Vision*. Springer, 2012. 1

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. 3

[17] Angjoo Kanazawa, David W Jacobs, and Manmohan Chandraker. Warpnet: Weakly supervised matching for single-view reconstruction. In *Proceedings of the IEEE Conference on CVPR*, 2016. 1, 2

[18] Animesh Karnewar, Oliver Wang, and Raghu Sesha Iyengar. Msg-gan: Multi-scale gradient gan for stable image synthesis. *CoRR*, 2019. 2

[19] Stefan Klein, Marius Staring, Keelin Murphy, Max A Viergever, and Josien PW Pluim. Elastix: a toolbox for intensity-based medical image registration. *IEEE transactions on medical imaging*, 29(1):196–205, 2009. 1

[20] Shun Miao, Sebastien Piat, Peter Fischer, Ahmet Tuysuzoglu, Philip Mewes, Tommaso Mansi, and Rui Liao. Dilated fcn for multi-agent 2d/3d medical image registration. In *32nd AAAI Conference*, 2018. 1, 2

[21] Gregory H Moore. The evolution of the concept of homeomorphism. *Historia Mathematica*, 34(3):333–343, 2007. 4

[22] Peter FM Nacken. Chamfer metrics in mathematical morphology. *Journal of Mathematical Imaging and Vision*, 4(3):233–253, 1994. 2

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015. 8

[24] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics (TOG)*, volume 23, pages 309–314. ACM, 2004. 8

[25] Siddharth Saxena and Rajeev Kumar Singh. A survey of recent and classical image registration methods. *International journal of signal processing, image processing and pattern recognition*, 7(4):167–176, 2014. 1

[26] Deqing Sun, Stefan Roth, JP Lewis, and Michael J Black. Learning optical flow. In *European Conference on Computer Vision*, pages 83–97. Springer, 2008. 5

[27] Zhiding Yu, Chen Feng, Ming-Yu Liu, and Srikumar Ramalingam. Casenet: Deep category-aware semantic edge detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7, 8

[28] Zhiding Yu, Weiyang Liu, Yang Zou, Chen Feng, Srikumar Ramalingam, BVK Vijaya Kumar, and Jan Kautz. Simultaneous edge alignment and learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 388–404, 2018. 1

[29] Aleksandar Zlateski, Ronnachai Jaroensri, Prafull Sharma, and Frédo Durand. On the importance of label quality for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1479–1487, 2018. 8