CVNet: Contour Vibration Network for Building Extraction

Ziqiang Xu[1], Chunyan Xu[1], Zhen Cui[1]*, Xiangwei Zheng[2], Jian Yang[1]
[1] School of Computer Science and Engineering, Nanjing University of Science and Technology
[2] School of Information Science and Engineering, Shandong Normal University
{xuziqiang, cyx, zhen.cui, csjyang}@njust.edu.cn, xwzhengcn@163.com

Abstract

The classic active contour model raises a great promising solution to polygon-based object extraction with the progress of deep learning recently. Inspired by the physical vibration theory, we propose a contour vibration network (CVNet) for automatic building boundary delineation. Different from the previous contour models, the CVNet originally roots in the force and motion principle of contour string. Through the infinitesimal analysis and Newton's second law, we derive the spatial-temporal contour vibration model of object shapes, which is mathematically reduced to second-order differential equation. To concretize the dynamic model, we transform the vibration model into the space of image features, and reparameterize the equation coefficients as the learnable state from feature domain. The contour changes are finally evolved in a progressive mode through the computation of contour vibration equation. Both the polygon contour evolution and the model optimization are modulated to form a close-looping end-to-end network. Comprehensive experiments on three datasets demonstrate the effectiveness and superiority of our CVNet over other baselines and state-of-the-art methods for the polygon-based building extraction. The code is available at https://github.com/xzq-njust/CVNet.

1. Introduction

Automatic building footprint extraction plays an important role in various higher level geographic and environmental applications, such as disaster assessment and rescuing [21], 3D-city modeling [13, 23], urban changing detection [27], earth observation and cartography [25] and so on. Most advanced object extraction methods [11, 26] fall into the category of pixel-wise segmentation, especially when CNNs have become the cornerstone in the pixel-wise image segmentation. Although the pixel-wise building extraction methods [11,26] generally perform well, most of them often lead to either big inadvertent fusion of adjacent instances or small scattered islands. Moreover, building borders are difficult to delineate with precise structures, and usually complex post-processing would be adopted to generate smooth shapes. At the same time, the segmentation methods store their results as raster data, which would result into high-memory requirement and lack the flexibility of distortion and zooming.

Instead, active contour model has the ability to address the aforementioned problems. Give an initialized polygon, a snake [12] would gradually converge to object boundary under the driven of both internal and external energies. To address the initialization and poor convergence to boundary concavities, Gradient Vector Flow [29] introduces a new external force for active contours, which can analyze it from a view of force balance though depending on energy functional. To boost the accuracy of building contour detection, recently, the modern CNN-based architectures have also been deployed to extract feature representation of building in [7, 19]. In particular, DSAC [19] integrate the powerful learning network with active contour model, which largely improves border delineation compared to the pixel-wise segmentation. Although a set of priors such as boundary continuity and smoothness may be adopted, self-intersections occur yet because minimizing energy cannot take the sequence of points into account. To overcome self-intersection, DARNet [7] takes polar coordinates to represent active contours, and deforms the contour points to the intersection of boundary and rays in the certain direction. In recent work [9], the authors proposed TDAC based on Level-Set model which estimates contours by pixel-wise prediction like image segmentation. ACDRNet [8] belongs to a general data-driven method that only considers the final goal of contours matching by directly optimize the final object contours and their masks. In contrast to the pixel-wise segmentation, these contour-based methods usually perform better in detecting various building contours, but they are limited in the energy principle of the active contour model itself [7, 9, 19], and yet the theory/principle of contour modeling is an open question.

*The corresponding author.

In a new perspective of physical vibration theory, we propose a novel contour vibration network (CVNet) to deal with automatic building boundary delineation. Intuitively, the evolution of object contour is just like the shape vibration (from an initial state to a terminal equilibrium state) of an elastic string (or rubber band) under some forces. The shape evolution could be modeled with physical motion equation, and the observed texture (or feature) patterns implicitly act one force to push or pull shape points. This case well matches the physical vibration theory, thus we introduce and adapt it to active contour model.

Different from the previous contour-based methods [7–9, 19], the work principle of CVNet conforms to Newton's second law of motion according to the force analysis of an infinitesimal string. Based on the force and motion principle of string vibration, the motion of contour is connected to internal/external forces of string, which are driven by the characteristics of image/object itself. By performing the infinitesimal analysis on the contour string, we build a spatial-temporal contour vibration model, which is mathematically reduced to second-order differential equation. Further, we concretize the vibration model in the space of image features, and take the reparameterizing trick of the equation coefficients. Thus the contour vibration equation is dynamic with parameterized coefficients to be learnt from the current contour state. The contour changes are finally evolved in a progressive mode through a recursive computation on contour vibration equation. Both the polygon contour evolution and the model learning are encapsulated into an end-to-end close-looping network framework. To verify the proposed method, we conduct extensive experiments on three building extraction datasets, including Vaihingen [20], Bing Huts [19] and our built large-scale Inria-building dataset. The experiments validate the feasibility that takes physical vibration theory for contour extraction, and the experiment results also demonstrate that our CVNet is effective and can achieve the state-of-the-art performance for the polygon-based building contour extraction.

In summary, our main contributions are three folds: i) propose a novel parametric-based ACM, named contour vibration model, inspired by the spirit of string vibration theory in physics; ii) design contour vibration network to dynamically evolve shapes based on mechanical equations of motion; iii) experimentally validate the feasibility and effectiveness in the polygon building contour extraction.

2. Related Work

Contour-based methods: With the development of deep learning and the accessibility of vast training data, the ideas based on contours were employed to treat the object segmentation problem, and an end-to-end network was optimized to predict the polygons which outline the object instances. For example, Castrejon et al. [3] proposed a Polygon-RNN to sequentially produce the polygonal annotation of the object inside the box. Polygon-RNN++ [1] was then proposed to accurately annotate high-resolution objects in images by employing reinforcement learning and Graph Neural Network. An end-to-end Curve-GCN framework [16] was also proposed to simultaneously predict all vertices using a Graph Convolutional Network. Above-mentioned works could be categorized as data-driven methods. Differently, Active Contour Models(ACMs) are usually built on forces or energies, which mainly fall into two classes: parametric based [7, 12, 19, 29] and geometric based [4, 9]. The parametric based snakes [12], which were first proposed to localize the boundaries of objects in 1988, could explicitly move predefined snake points by minimizing an energy. An external force for active contours [29], named gradient vector flow (GVF), was then proposed to address the initialization and poor convergence to object boundaries. The geometric based method [4] proposed a model to detect objects by evolving the implicit function.

Building extraction: Object extraction from the aerial imagery has gradually become a significant research topic in the field of remote sensing and computer vision [2, 7, 14, 15, 28]. For example, a graph-cycle based object localization algorithm [24] was proposed to the polygonal object detection, where an efficient graph-partition search algorithm was defined by using the cyclomatic number and these object contours were extracted by preserving these nodes and edges with a high weight. As in [26], the building contour extraction task can be addressed with these classic semantic segmentation networks [5, 6, 17]. Kaiser et al. [11] adapted a CNN framework for semantic segmentation of building and road in aerial images which showed that weakly labeled training data significantly improved the segmentation performance. By employing ACMs, DSAC framework [19] integrated priors and constraints (e.g., continuous boundaries, smooth edges, and sharp corners) into the process of building instance segmentation. Instead of parameterizing the contour using Euclidean coordinates, Cheng et al. [7] adopted polar coordinates to evolve a polygon-based contour of building, and then proposed DARNet for automatic building segmentation in an end-to-end fashion. In order to integrate the advantages of CNNs and ACMs, Hatamizadeh et al. [9] proposed trainable deep active contours to deal with the automated segmentation of buildings in remote sensing imagery. Gur et al. [8] proposed an ACDRNet to shift a contour based on a 2-channel displacement field, which was optimized by employing both the polygon shape and the segmentation mask. Different from these previous active contours, we start from the physical vibration theory to drive the spatial-temporal contour vibration model of object/building shapes, which should be the first time to our knowledge in the object/building boundary extraction problem.

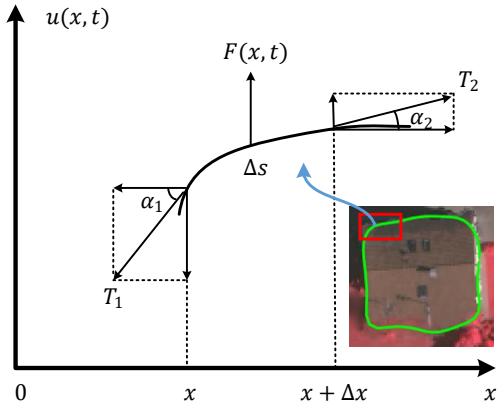

Figure 1. Infinitesimal analysis of contour vibration.

3. Methods

In this section, the writting organization takes a gradual way. We first introduce the basic physical principle behind the method and then build the string vibration model for the building contour extraction, finally derive the contour evolution process.

3.1. Contour Vibration Principle

In the most learning-based algorithms of contour detection, object contours are usually modeled in a gradual string evolution process, and finally reach an optimal/approximate stable state. As an intrinsic revelation in physics, the contour evolution falls into the theory category of wave equation, analogical to many real-world physical phenomenons, such as vibration strings, water waves, sound waves, and so on. In order to better understand and derive the contour evolution process, we exploit the infinitesimal method to analyze the string vibration principle of a small arc, as shown in Fig. 1.

In the figure, the horizontal x-axis denotes the positions of contour string, and the vertical axis $u(x,t)$ represents the displacement from equilibrium state (also real contour) at time t. Since the string is soft, tight and uniform, we may think internal tensions of any positions are equal numerically in their magnitudes. Now we dissect the force situation of an infinitesimal Δs at the position x. The infinitesimal Δs is not only affected by the internal tension, but also suffers from an additional pulling or resistance force F (w.r.t positive or negative values). The internal tension contains two parts, the left-direction tension $\mathbf{T}_1$ and right-direction tension $\mathbf{T}_2$, which have the same magnitude but different directions, i.e., $T_1 = T_2 = T$. Note the two tensions are corresponding to tangent vectors of the string. The external force comes from additional driving or resisting strength. Given a total external force F, the infinitesimal has the force $F(x,t)\Delta x$ at the position x and the time t.

According to the Newton's second law, we can obtain the equation of motion for the infinitesimal Δs in the direction u:

$$T_2 \sin\alpha_2 - T_1 \sin\alpha_1 + F\Delta x = m\frac{\partial^2 u}{\partial t^2}, \tag{1}$$

where α_1, α_2 are the angles between tangent vectors (at the ends of Δs) and x-axis, m is the mass of the string Δs, and $\frac{\partial^2 u}{\partial t^2}$ denotes the acceleration of the string vibration. Suppose the density of string is ρ, then we have $m = \rho\Delta x$. As string vibration is usually small, i.e., $\alpha_1, \alpha_2 \to 0$, we can make the following approximation,

$$\sin\alpha_1 \approx \tan\alpha_1 = \left.\frac{\partial u}{\partial x}\right|_x, \tag{2}$$

$$\sin\alpha_2 \approx \tan\alpha_2 = \left.\frac{\partial u}{\partial x}\right|_{x+\Delta x}. \tag{3}$$

The external force may come from driving or resisting strength. Here we take the resistance force to analyze the string vibration. Obviously, when the resistance force is negative, it may be viewed as pulling force. According to the physical law, the resistance force is proportional to the velocity of the object, formally,

$$F(x,t) \doteq -k\frac{\partial u}{\partial t}, \tag{4}$$

where k is the damping factor, and the minus sign indicates an opposite direction. The additional force depends on the motion situation of the string. The higher the speed, the larger the resistance force. It means that we expect to suppress the situation that the contour vibrates dramatically, which would lead to the learning smoothness. According to Eqns. (2), (3) and (4), we can rewrite Eqn. (1) as follows,

$$\begin{aligned}
\rho\frac{\partial^2 u}{\partial t^2}\Delta x &= T(\sin\alpha_2 - \sin\alpha_1) + F\Delta x \\
&= T(\tan\alpha_2 - \tan\alpha_1) + F\Delta x \\
&= T\left(\left.\frac{\partial u}{\partial x}\right|_{x+\Delta x} - \left.\frac{\partial u}{\partial x}\right|_x\right) + F\Delta x \\
&= T\Delta\left(\frac{\partial u}{\partial x}\right) + F\Delta x \\
&= T\frac{\Delta\left(\frac{\partial u}{\partial x}\right)}{\Delta x}\Delta x + F\Delta x \\
&\approx T\frac{\partial^2 u}{\partial x^2}\Delta x + F\Delta x \\
&= T\frac{\partial^2 u}{\partial x^2}\Delta x - k\frac{\partial u}{\partial t}\Delta x.
\end{aligned} \tag{5}$$

In the above calculation of Eqn. (5), we use the differential definition, $\frac{\partial^2 u}{\partial x^2}|_x = \lim_{\Delta x \to 0} \frac{\frac{\partial u}{\partial x}|_{x+\Delta x} - \frac{\partial u}{\partial x}|_x}{\Delta x}$. Eliminating the common term Δx, we can obtain the string vibration equation as

$$\frac{\partial^2 u}{\partial t^2} - a^2\frac{\partial^2 u}{\partial x^2} + 2b\frac{\partial u}{\partial t} = 0, \tag{6}$$

where $a = \sqrt{T/\rho}$, $2b = k/\rho$. In the above equation, a, b are the meaningful physical parameters. In other words, the parameters depend on the concrete physical systems if we generalize the vibration equation in Eqn. (6) to a universal case, as described in the following section.

3.2. Contour Vibration Model

The vibration equation in Eqn. (6) basically describes the vibration rule of shape contours. The crucial problem is to determine the proper parameters $\{a, b\}$, which should be derived from the dependent system. To address this problem, we reparameterize the physical parameters $\{a, b\}$ of contour vibration equation as $\{\alpha(x), \beta(x)\}$[1], which changes dynamically with the input state x. Thus, the dynamic model could be established as follows,

$$\frac{\partial^2 u}{\partial t^2} - \alpha(x)\frac{\partial^2 u}{\partial x^2} + \beta(x)\frac{\partial u}{\partial t} = 0. \tag{7}$$

It means that the contour vibration should comply with the above equation rule. The aim is to learn those dynamic parameters. Once they are solved, we can infer the contour evolution process. Below we concretize the above vibration rule into the contour variation model of image.

Given an arbitrary vertex p on the contour boundary of an image object, its position changes under the effects of the internal and external forces as analyzed in Section 3.1. In the image space, these forces may be understood as the role of the observed features (e.g., convolution features), which will pull or resist contour variations. We next introduce how to concretize each term of the model in Eqn. (7) in the discrete image space:

- Internal Term $\frac{\partial^2 u}{\partial x^2}$: The second-order derivative term w.r.t x is defined upon the internal tension as described in Eqn. (1) and (5). In other words, the internal tension exists between contour points to perform the mutual constraint. The front coefficient $\alpha(x)$ controls the degree of internal tension. In a discrete space, we define the internal term $\frac{\partial^2 u}{\partial x^2}$ as
$$\left.\frac{\partial^2 u}{\partial x^2}\right|_p \doteq u_{p+1} + u_{p-1} - 2u_p, \tag{8}$$
where p is one vertex of the contour.
- External term $\frac{\partial u}{\partial t}$: The first-order derivation term w.r.t t comes from the external force as described in Eqn. (4), which makes the contour curve vibrate in the temporal space. The coefficient $\beta(x)$ encodes the expansion or resistance force when taking positive or negative values. Accordingly, we define the discrete numerical operation as
$$\frac{\partial u}{\partial t} \doteq \frac{u_t - u_{t-1}}{\Delta t}. \tag{9}$$

[1]In the following section, we often omit x and directly use α, β due to the clear context.

- Accelerated speed term $\frac{\partial^2 u}{\partial t^2}$: The second-order derivation term w.r.t t comes from the motion equation in Eqn. (1), which reflects the acceleration of contour changes. We concretize the term as
$$\frac{\partial^2 u}{\partial t^2} \doteq \frac{u_{t+1} + u_{t-1} - 2u_t}{(\Delta t)^2}. \tag{10}$$

After substituting the terms of Eqn. (7) with the above three discrete equations, we can obtain the final contour vibration model,

$$\left[\frac{u_{t+1} + u_{t-1} - 2u_t}{(\Delta t)^2}\right]_p - [\alpha_{p+1}u_{t,p+1} + \alpha_{p-1}u_{t,p-1} - 2\alpha_p u_{t,p}] + \beta_p\left[\frac{u_t - u_{t-1}}{\Delta t}\right]_p = 0, \tag{11}$$

where $u_{t,p}$ denotes the contour point information of vertex p at the time t, and $\alpha_p = [\alpha(x)]_p, \beta_p = [\beta(x)]_p$ with the abbreviation. If we sample n points from the object contour shape, each point $p \in \{1, 2, \cdots, n\}$ should conform to this equation, where $\alpha(x), \beta(x)$ need to be learnt, e.g., using neural network in Section 3.4.

3.3. Contour Evolution Computation

To detect the contour of object, we need to solve u in the above contour vibration model of Eqn. (11). To this end, below we derive the contour evolution process along the time axis. For n sampling points, we denote the object contour with a vector of n vertices, i.e.,

$$\mathbf{u}_t = [u_{t,1}, u_{t,2}, ..., u_{t,n}]^\top, \tag{12}$$

where $\top$ is the transpose of vector/matrix. We can rewrite the above equation set, described in Eqn. (11) with $p = 1, 2, \cdots n$, as the matrix formula,

$$\mathbf{u}_{t+1} + \mathbf{u}_{t-1} - 2\mathbf{u}_t - \mathbf{A}\mathbf{u}_t(\Delta t)^2 + \boldsymbol{b} \odot (\mathbf{u}_t - \mathbf{u}_{t-1})\Delta t = 0, \tag{13}$$

where $\boldsymbol{b} = [\beta_1, \beta_2, \cdots, \beta_n]^\top \in \mathbb{R}^n$, $\odot$ denotes the element-wise multiplication, and the matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ takes the use way of Matlab as

$$\mathbf{A} = \begin{bmatrix} -2\alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \alpha_1 & -2\alpha_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1 & \cdots & \alpha_{n-1} & -2\alpha_n \end{bmatrix},$$

For Eqn. (13), we can derive the recursive formula as

$$\mathbf{u}_{t+1} = 2\mathbf{u}_t - \mathbf{u}_{t-1} + \mathbf{A}\mathbf{u}_t(\Delta t)^2 - \boldsymbol{b} \odot (\mathbf{u}_t - \mathbf{u}_{t-1})\Delta t, \tag{14}$$

It indicates that the contour at the current moment $t + 1$ could be recursively computed from the states of the previous moments and the estimated parameters $\{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$.

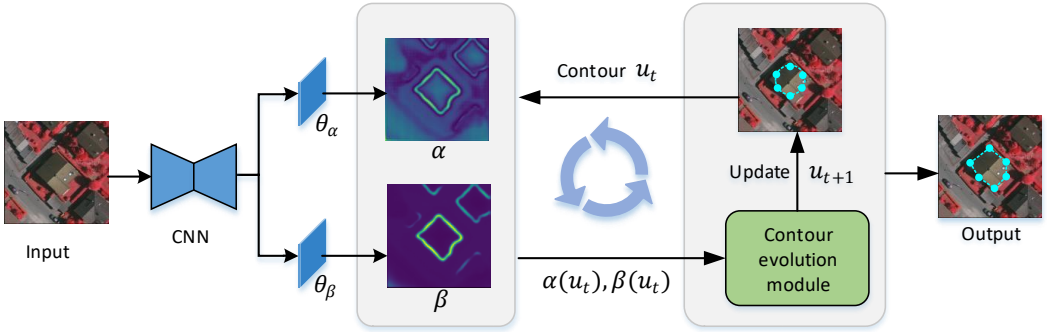

Figure 2. An illustration of CVNet. We use CNN to extract feature $\mathbf{F}$ and then derive vibration parameters $\{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$ through two separate network branches. According to the current contour $\mathbf{u}_t$, we can obtain the parameters of dynamic equation by indexing $\boldsymbol{\alpha}, \boldsymbol{\beta}$ maps. The object contour is refined with the iterative evolution.

3.4. Contour Learning Network

Network Architecture: According to the above analysis, we design an end-to-end network to fulfill contour evolution $\mathbf{u}_t$ and parameter learning $\{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$, as shown in Fig. 2. Motivated by the powerful representation of CNN, we employ it to extract features from an input image. The feature extraction network may be chosen from those conventional convolutional networks. After encoding the input image, we can obtain the convolution feature $\mathbf{F}$, which is used for the downstream contour learning.

We derive the vibration parameters $\{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$ by feeding the encoded feature $\mathbf{F}$ into two separate network-branches (here using one convolutional layer), whose parameters are denoted as $\theta_\alpha, \theta_\beta$ respectively. In other word, given a vertex p on the contour $\mathbf{u}$, we can estimate $\alpha_p = f(\mathbf{F}_p, \theta_\alpha), \beta_p = f(\mathbf{F}_p, \theta_\beta)$, where $\mathbf{F}_p$ is the feature of vertex p, f is the convolutional network with parameter θ to be learnt. To simplify the repetitive parameter estimation process, we can produce an entire parameter map of $\{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$. According to the current contour $\mathbf{u}_t$ of object, we can directly index the parameter $\boldsymbol{\alpha}(\mathbf{u}_t), \boldsymbol{\beta}(\mathbf{u}_t)$ by using the bilinear interpolation operation. To better accelerate the evolution, a good initial contour is required. Here we follow the same strategy [7] to assign the initial object contour $\mathbf{u}_0$.

Based on the estimated vibration parameters $\{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$ and the previous contours (i.e., $\mathbf{u}_t$ and $\mathbf{u}_{t-1}$), we can infer the next contour $\mathbf{u}_{t+1}$ as defined in the formula (14), also named contour evolution module in Fig. 2. Finally, the object contour is refined with iterative evolution.

Learning and Inference: In order to match the estimated polygon $\widehat{\mathcal{S}}$ with the ground truth polygon $\mathcal{S}$, we use the symmetric Chamfer Distance as the loss function. Both $\widehat{\mathcal{S}}$ and $\mathcal{S}$ are the sets of coordinates of contour vertices, but we are unclear about the ordering of contour vertices. The previous methods (such as DARNet [7]) need an explicit ordering, such that anyone vertex moves in a fixed direction and corresponds to the target position. To bypass the explicit matching process between two vertex sets $\widehat{\mathcal{S}}$ and $\mathcal{S}$, we use the Chamfer Distance loss, which is defined as follows,

$$\zeta(\widehat{\mathcal{S}}, \mathcal{S}) \doteq \sum_{\widehat{u} \in \widehat{\mathcal{S}}} \min_{u \in \mathcal{S}} \|\widehat{u} - u\|_2^2 + \sum_{u \in \mathcal{S}} \min_{\widehat{u} \in \widehat{\mathcal{S}}} \|u - \widehat{u}\|_2^2 . \quad (15)$$

When taking the batch process, we can accumulate the loss of all samples during the training. Due to the advantage of not caring about the order of points on the contour, the Chamfer Distance loss can move the vertices to the target boundaries as soon as possible. Furthermore, the symmetric structure prevents multiple points on a polygon from converging on the same point.

To train the model in an end-to-end manner and back-propagate gradients to the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ parameters, we take the point coordinate as floating point number and make use of bilinear interpolation to acquire the value of the map at a point, as used in spatial transformer networks [10]. In the test stage, we can follow the process as plotted in Fig. 2, where the vibration parameters $\{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$ rely on the input and thus are updated dynamically with better flexibility.

3.5. Theoretical Analysis

The existing active contour models only seek for an appropriate contour to minimize the energy functional, but do not consider how contours change at the starting point. Our method not only optimizes an equilibrium state of contour evolution, but also models the dynamic motion of contours. They could be demonstrated by Eqn.(7) in our paper vs Eqn.(4) in DARNet [7] or Eqn.(1) in DSAC [19] or Eqn.(3) in TDAC [9]. Under a reasonable assumption of physical motion rule, the shape space could be well-constrained with some flexibility, which naturally reduces shape shifting during contour evolution.

4. Experiments

4.1. Experimental setup

Datasets: To evaluate our proposed CVNet, we conduct comprehensive experiments on two public aerial datasets (namely Vaihingen [20], Bing Huts [19]), and one new

Table 1. Comparison of building extraction performances on the Vaihingen and Bing Huts datasets.

Method		Backbone	Vaihingen					Bing Huts				
			mIoU	WCov	BoundF	Dice	Time(ms)	mIoU	WCov	BoundF	Dice	Time(ms)
Pixel-level segmentation	FCN	ResNet	75.60	77.50	38.30	84.20	–	68.40	76.14	39.19	79.90	–
	FCN	UNet	78.60	81.80	40.20	87.40	–	64.90	75.70	41.27	77.20	–
	FCN	DSAC	81.00	81.40	64.60	–	–	69.80	73.60	30.30	–	–
	FCN	DARNet	87.20	86.80	76.80	–	–	74.50	77.50	37.70	–	–
	ACDRNet	–	90.33	**90.62**	78.75	94.81	–	75.53	76.12	36.81	85.44	–
Geometric based ACM	TDAC-const λs	–	83.79	82.70	73.21	91.18	–	73.02	74.21	48.25	84.53	–
	TDAC	–	89.16	90.54	78.12	94.26	–	80.39	81.05	**53.50**	**89.12**	–
Parametric based ACM	DSAC	–	71.10	70.76	36.44	–	102.96	38.74	44.61	37.16	–	68.78
	DSAC	DARNet	60.37	61.12	24.34	–	–	57.23	63.09	15.98	–	–
	DARNet	–	88.20	88.10	75.90	93.66	130.29	75.20	77.00	38.00	85.21	104.77
	Ours	DARNet	**90.43**	90.46	**81.76**	**94.91**	**33.12**	**80.42**	**82.26**	46.37	88.74	**31.0**

constructed Inria-building dataset. The Vaihingen building dataset [20] consists primarily of buildings in a German city. The original images are 512 $\times$ 512 at a resolution of 9 cm/pixel. There are 168 buildings in total, which are divided into 100/68 examples for training and testing, respectively. All images contain centered buildings with a highly complex environment, which makes the task challenging. The Bing huts dataset [19] consists of huts located in a rural area of Tanzania. There are 606 images in total with an original size of 64 $\times$ 64 at a resolution of 30 cm/pixel. We use 335 samples to train the models and the remaining 271 samples as a test set. This dataset is more challenging due to the lower spatial resolution and low contrast that are exhibited in the images. As we cannot obtain the large-scale Torontocity dataset for evaluating the effectiveness of our CVNet, we construct a new large building extraction dataset, named Inria-building, whose samples are cropped from Inria Aerial Image Labeling Dataset [18]. The building images are with a wide range of urban settlement appearances, from different geographic locations. There are 18,952 building images in total, which are with 128 $\times$ 128 pixels at a spatial resolution of 0.3m. We divide data into train, validation and test sets with the ratio of 6:2:2. The constructed Inria-building dataset exhibits the following distinctive characteristics: (a) a large number of image samples with high spatial resolution, (b) a wide range of urban settlement including 5 cities, (c) many aerial images occluded by trees but labeled in the ground-truth, (d) the positions of building objects are diverse, where the building objects may not locate in the center of images. Compared to other aerial datasets for contour extraction, Inria-building dataset is more diverse, comprehensive and challenging.

Implementation details: We use DRN network [30] to encode the inputs, and then employ three transposed convolutional layers for learning the vibration parameters. We train CVNet using stochastic gradient descent with a batch size of 10 images, momentum of 0.3, and weight decay of 1e-5. The learning rate is initialized at 0.008 and divided by 2 every 50 epochs while 250 epochs in total. Throughout the whole training process, we also implement data augmentation including random flip, scale and rotation. We discretize our building contour with 60 points, and set the radius value of initialized contour as 20 pixels for Vaihingen and 12 pixels for Bing Huts. Followed the same evaluation metric in [9], we also utilize four different metrics, including Dice, mean Intersection over Union (mIoU), Weighted Coverage (WCov) [22] and Boundary F score (BoundF) [7]. All models in our experiment are trained and tested based on the PyTorch platform on a single NVIDIA 2080Ti GPU.

Table 2. Comparison of building extraction performances on the Inria-building dataset.

Method	Inria-building dataset			
	mIoU	WCov	BoundF	Dice
DSAC	35.1	37.78	5.85	51.18
DARNet	65.83	60.45	33.04	77.21
Ours	**77.61**	**75.63**	**42.20**	**86.73**

4.2. Comparisons with state-of-the-arts

We compare our proposed CVNet with several state-of-the-art building extraction approaches [7, 19] on Vaihingen and Bing Huts datasets. Table 1 reports the performance of our CVNet model and comparisons with these geometric-/parametric-based ACMs and these pixel-level segmentation methods on overall metrics. The metric BoundF indicates the similarity of geometrical shape between prediction and ground-truth. Specifically, our CVNet can significantly outperform these baselines in terms of BoundF on Vaihingen building dataset. The result of 3% higher than all methods demonstrates that our model has a better boundary quality. Correspondingly, metrics mIoU/WCov/Dice measure the overlap ratio of regions and our method achieves state-of-the-art performances. In terms of mIoU score, we outperform TDAC by 1.27% on Vaihingen and ACDRNet by 4.89% on Bing Huts respectively. In conclusion, our CVNet performs well not only at the region, but also at the

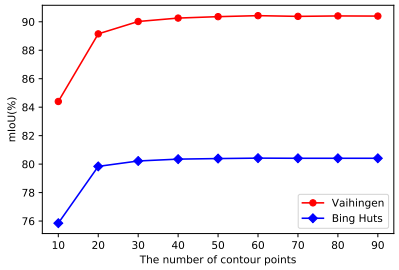

Figure 3. Performance comparisons with different number of contour points.

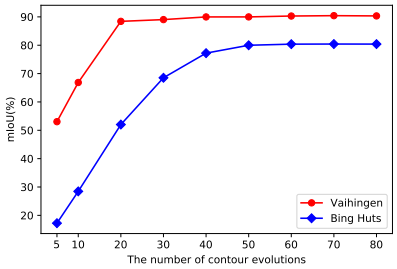

Figure 4. Performance comparisons with different number of contour evolutions.

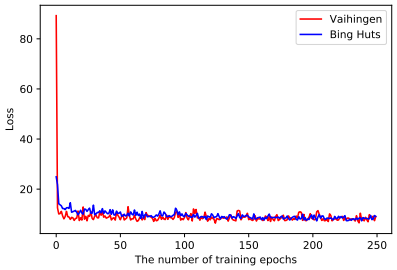

Figure 5. The loss curves with different number of epochs.

boundary.

In principle, our method models second-order information w.r.t time while DARNet actually models one-order function w.r.t time. Hence, at the stage of contour evolution, our method needs less iterative times than DARNet for well solutions. In experiment, we observe that 70 steps are enough for our CVNet while DARNet performs 200-step evolution for well convergence. To be more clear, we summarize the speed comparisons between our CVNet, DSAC and DARNet on the two datasets in Table 1, where the last column refers to the inference time of a single image. Our CVNet outperforms these classic DSAC and DARNet in both speed and accuracy.

Table 2 further reports the performance comparisons between our CVNet model with other baselines (including DSAC [19] and DARNet [7]) on our constructed Inria-building dataset. We achieve the best performance on overall metrics. This indicates that our CVNet method performs very well even on a large-scale challenging building extraction dataset in the remote sensing imagery.

4.3. Ablation study

Influence of different contour points: We explore the building extraction performance of our CVNet method with different number of contour points, and Fig. 3 shows the detailed mIoU results on Vaihingen and Bing Huts datasets. When the number of contour points increases, the performances of our CVNet are significantly improved. When we keep increasing the number after 60 contour points, the building extraction results would be slightly deteriorated. The reason is that, the building object with less points would has difficulty to regress its optimal contour, while the building object with more points will add the complexity of building extraction, which will weaken our model.

Influence of different contour evolutions: The building contour will be iteratively evolved/updated in our CVNet framework. We report the building extraction performances with different number of contour evolutions on Vaihingen and Bing Huts datasets, as can be shown in Fig. 4. The performances are increased as the number of evolutions increases, while they are with a slow descent with the CVNet learning process after 70 evolutions. It demonstrates that the learned contour process from the training samples can be optimized to improve the discriminative capability of CVNet model, and gradually reach a stable state of the building contour.

Stability analysis: For saving the computation cost, the loss is computed after multiple iterations on contour evolution, and then the gradients can be back-propagated to optimize network parameters. Fig. 5 shows the loss curves with different number of epochs on the Vaihingen and Bing Huts datasets. The losses are rapidly decreasing as the number of epochs increases from the beginning to 5 epochs, while with a slow descent later. After learning with a few epochs, the CVNet would become more stable with the dynamical update of the two parameters $\{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$, and then further improve the performance of building contour extraction.

Qualitative discussion: We present the qualitative comparison of our CVNet against two state-of-the-art baselines (i.e., DSAC [19], DARNet [7]) on the Vaihingen, Bing Huts, and our built Inria-building datasets. As illustrated in Fig. 6 (a)-(d), the existing DARNet and DSAC models have difficulty coping with the topological changes of the buildings and fail to appropriately capture sharp edges, but our CVNet performs well when predicting the contours of building objects in the aerial images even with the complex background. This demonstrates that the CVNet model is beneficial to predict the effective boundaries of building objects and generate more precise prediction in challenging aerial images. For better understanding the dynamic process of contour evolution, we also show the visual maps of vibration parameters (i.e., $\boldsymbol{\alpha}, \boldsymbol{\beta}$) in Fig. 6 (e) and Fig. 6 (f). We further present the predicted results of building contours with increasing the number of evolutions on the Vaihingen dataset, as illustrated in Fig. 7. It clearly demonstrates the effectiveness of our proposed CVNet to cope with the active building contour learning problem.

5. Conclusion

In this work, we proposed a novel CVNet architecture for automatic building extraction in the aerial imagery, which is the first time to introduce the physical Vibrating String

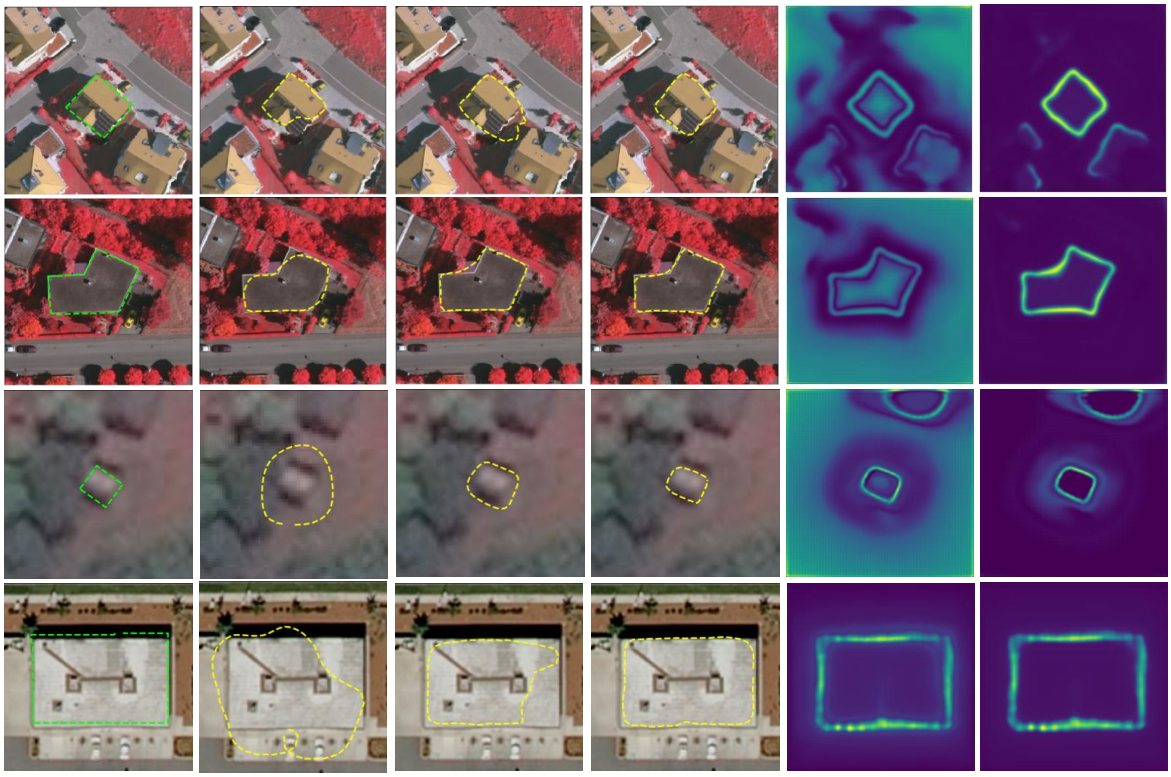

Figure 6. Comparative visualization of the labeled image and the predicted results of DSAC, DARNet, and our CVNet for the Vaihingen (the top two rows), Bing Huts (the third row), and Inria-building (the below row) datasets.

Figure 7. Visualization of the contour evolution process with different number of evolutions for the Vaihingen dataset.

Principle into the object extraction/segmentation task to our knowledge. Given an input image, we extract deep feature representations through a classic convolutional neural network, which can be further utilized to iteratively guide the prediction process of building contour. The contour changes have been evolved in a progressive mode through the computation of contour vibration equation. Both the polygon contour evolution and the model optimization can be modulated to form a close-looping end-to-end network. Extensive experimental results clearly demonstrated the effectiveness of the proposed CVNet in the automatic building extraction task.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (Grants Nos. 62072244, 61972204), the fundamental research funds for the central universities (No. 30921011104), the Natural Science Foundation of Shandong Province (No. ZR2020LZH008), and partly collaborated with State Key Laboratory of High-end Server & Storage Technology.

References

[1] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *CVPR*, pages 859–868, 2018. 2

[2] Rheannon Brooks, Trisalyn Nelson, Krista Amolins, and G Brent Hall. Semi-automated building footprint extraction from orthophotos. *Geomatica*, 69(2):231–244, 2015. 2

[3] Lluis Castrejon, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Annotating object instances with a polygon-rnn. In *CVPR*, pages 5230–5238, 2017. 2

[4] T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001. 2

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 2

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018. 2

[7] Dominic Cheng, Renjie Liao, Sanja Fidler, and Raquel Urtasun. Darnet: Deep active ray network for building segmentation. In *CVPR*, pages 7431–7439, 2019. 1, 2, 5, 6, 7

[8] Shir Gur, Tal Shaharabany, and Lior Wolf. End to end trainable active contours via differentiable rendering. In *ICLR*, 2020. 1, 2

[9] Ali Hatamizadeh, Debleena Sengupta, and Demetri Terzopoulos. End-to-end trainable deep active contour models for automated image segmentation: Delineating buildings in aerial imagery. In *ECCV*, pages 730–746, 2020. 1, 2, 5, 6

[10] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *NeurIPS*, volume 28, 2015. 5

[11] Pascal Kaiser, Jan Dirk Wegner, Aurélien Lucchi, Martin Jaggi, Thomas Hofmann, and Konrad Schindler. Learning aerial image segmentation from online maps. *IEEE Transactions on Geoscience and Remote Sensing*, 55(11):6054–6068, 2017. 1, 2

[12] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988. 1, 2

[13] Florent Lafarge, Xavier Descombes, Josiane Zerubia, and Marc Pierrot-Deseilligny. Automatic building extraction from dems using an object approach and application to the 3d-city modeling. *ISPRS Journal of photogrammetry and remote sensing*, 63(3):365–381, 2008. 1

[14] Chengzheng Li, Chunyan Xu, Zhen Cui, and et al. Feature-attentioned object detection in remote sensing imagery. In *ICIP*, pages 3886–3890, 2019. 2

[15] Chengzheng Li, Chunyan Xu, Zhen Cui, Dan Wang, and et al. Learning object-wise semantic representation for detection in remote sensing imagery. In *CVPRW*, June 2019. 2

[16] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast interactive object annotation with curve-gcn. In *CVPR*, pages 5257–5266, 2019. 2

[17] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 2

[18] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. In *IEEE International Geoscience and Remote Sensing Symposium*, pages 3226–3229, 2017. 6

[19] Diego Marcos, Devis Tuia, Benjamin Kellenberger, Lisa Zhang, Min Bai, Renjie Liao, and Raquel Urtasun. Learning deep structured active contours end-to-end. In *CVPR*, pages 8877–8885, 2018. 1, 2, 5, 6, 7

[20] Franz Rottensteiner, Gunho Sohn, Jaewook Jung, Markus Gerke, Caroline Baillard, Sbastien Bnitez, and U Breitkopf. International society for photogrammetry and remote sensing, 2d semantic labeling contest. *http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html*. 2, 5, 6

[21] Liora Sahar, Subrahmanyam Muthukumar, and Steven P French. Using aerial imagery and gis in automated building footprint extraction and shape recognition for earthquake risk assessment of urban inventories. *IEEE Transactions on Geoscience and Remote Sensing*, 48(9):3511–3520, 2010. 1

[22] Nathan Silberman, David Sontag, and Rob Fergus. Instance segmentation of indoor scenes using a coverage loss. In *ECCV*, pages 616–631, 2014. 6

[23] Helene Sportouche, Florence Tupin, and Leonard Denise. Building extraction and 3d reconstruction in urban areas from high-resolution optical and sar imagery. In *Joint Urban Remote Sensing Event*, pages 1–11, 2009. 1

[24] Xiaolu Sun, C Mario Christoudias, and Pascal Fua. Free-shape polygonal object localization. In *ECCV*, pages 317–332, 2014. 2

[25] Mustafa Turker and Dilek Koc-San. Building extraction from high-resolution optical spaceborne images using the integration of support vector machine (svm) classification, hough transformation and perceptual grouping. *International Journal of Applied Earth Observation and Geoinformation*, 34:58–69, 2015. 1

[26] Shenlong Wang, Min Bai, Gellert Mattyus, Hang Chu, Wenjie Luo, Bin Yang, Justin Liang, Joel Cheverie, Sanja Fidler, and Raquel Urtasun. Torontocity: Seeing the world with a million eyes. *arXiv preprint arXiv:1612.00423*, 2016. 1, 2

[27] Yanhua Xie, Anthea Weng, and Qihao Weng. Population estimation of urban residential communities using remotely sensed morphologic data. *IEEE Geoscience and Remote Sensing Letters*, 12(5):1111–1115, 2015. 1

[28] Chunyan Xu, Chengzheng Li, Zhen Cui, and et al. Hierarchical semantic propagation for object detection in remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 58(6):4353–4364, 2020. 2

[29] Chenyang Xu and Jerry L Prince. Snakes, shapes, and gradient vector flow. *IEEE TIP*, 7(3):359–369, 1998. 1, 2

[30] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *CVPR*, 2017. 6