

Article

# A New Bayesian Edge-Linking Algorithm Using Single-Target Tracking Techniques

Ji Won Yoon

Center for Information Security Technologies (CIST), Korea University, Seoul 02841, Korea; jiwon\_yoon@korea.ac.kr; Tel.: +82-2-3290-4886

Academic Editor: Angel Garrido

Received: 31 July 2016; Accepted: 16 November 2016; Published: 1 December 2016

**Abstract:** This paper proposes novel edge-linking algorithms capable of producing a set of edge segments from a binary edge map generated by a conventional edge-detection algorithm. These proposed algorithms transform the conventional edge-linking problem into a single-target tracking problem, which is a well-known problem in object tracking. The conversion of the problem enables us to apply sophisticated Bayesian inference to connect the edge points. We test our proposed approaches on real images that are corrupted with noise.

**Keywords:** boundary detection; edge linking; single-target tracking

---

## 1. Introduction

Edge detection is a key and fundamental step in many computer vision and image-processing applications since it produces useful features of raw images. Edge detection takes a gray-scale image as input and generates a binary image map that includes black-and-white values at each pixel. Many researchers have devoted their efforts to inventing edge-detection algorithms.

Edge detection is based on the detection of abrupt local changes in the image intensity. In practice, edge models are classified according to their intensity profiles: step, ramp and roof. Given the edge model, edge detection involves three fundamental steps: image smoothing for noise reduction, detection of edge points and edge localization [1]. That is, we first reduce the noise in images by using several low-pass filters [2,3] in the image smoothing for noise reduction step. The detection of edge points step is a local operation that extracts all points that are potential candidates for edge points. In other words, the algorithms consider all of the points and then use thresholds, a histogram, gradients and zero crossing to identify promising edge points [4–6]. The last step is edge localization, the objective of which is to select from the candidate edge points only those points that would be meaningful to build the desired features. However, it is not straightforward to obtain the complete set of desired features even after the above three steps. Although edge detection should yield sets of pixels that only coincide with edges, the actual underlying edges are seldom characterized by the detected pixels because of unwanted noise in raw images, spurious breaks in the edges and improperly-chosen threshold values. Therefore, edge-linking or boundary detection/tracing algorithms often follow edge detection to eventually recognize the simplified and salient object of interest from the image. Various algorithms have been developed to connect the detected edge points. Broken edges are extended along the directions of their respective slopes by using an (adaptive) morphological dilation operation [7,8]. Multiple resolution and hierarchical approaches to edge detection were also proposed by [9,10]. Another well-known approach for edge linking is the Hough transform [11]. The graph search problem with the  $A^*$  algorithm has also been used to link edges [12]. Probabilistic relaxation techniques have been applied to label edges for boundary tracing, as well [13–16]. A predictive edge-linking algorithm was also developed by considering the prediction generated by past movements [17]. In addition to these, many other approaches to developing edge-linking algorithms have been proposed [18–21]. However,

most algorithms are not effective for general-purpose usage, and they are highly dependent on the data and applications because of the presence of noise and long interruptions (the existence of occlusions).

Therefore, we propose a new approach to connect the edge points to create meaningful longer edges even in the presence of a large amount of noise, unwanted interruptions and long occlusions. Our idea to link the detected points is to combine our linking edges with data association in the object target tracking methodology. The data association technique in target tracking is commonly used to identify detected points of objects in a cluttered environment, and it classifies observations as being either target oriented or non-target oriented. That is, such object tracking schemes can be applied to classify the detected points into desired edges or unwanted clutter. In the work presented in this paper, the target tracking method for boundary determination is designed as part of a Bayesian scheme, and the posterior distribution is the objective function with the shape prior in a Markov random field (MRF).

The contributions of our paper are fourfold: (1) introducing a new probabilistic relaxation technique using single-target tracking techniques; (2) overcoming multiple occurrences of noise and long occlusions; (3) designing three efficient statistical methodologies; and (4) application to interactive segmentation.

This paper consists of several sections. Section 3 presents background knowledge related to the techniques that are used in this paper. After first explaining the intuitive idea of our approach in Section 2, various algorithms are proposed in Section 4. Afterwards, we conclude with our results and a related discussion.

## 2. Motivation (Idea)

In this paper, we propose a new approach to linking edges based on our idea to introduce single-target tracking algorithms. Target tracking algorithms have found use in many application domains of computer vision and image processing [22–26]. Given the raw image in Figure 1a, we have a binary image in Figure 1b after applying several detection algorithms, such as Canny and Sobel. Of course, advanced statistical detection algorithms can be applied [27,28]. Here, if we want to obtain the boundary of the triangle from the detected points, we can click any point inside of the polygon in an interactive way [29]. We name this point an origin. Then, we obtain  $T$  lines from the binary image in a radial way using Bresenham's line-drawing algorithm, as shown in Figure 1c. This is similar to the approach followed in the star convexity shape prior approach [30]. Each line can cross the detected points that are potential edge candidates, and we set the crossed points to be the measurements of our interest in the tracking system. That is, each line in Figure 1c is the same as the time slot in the tracking domain, as shown in Figure 1d. Now, the radial distance ( $r$ ) of the crossed points represents the value of the observations. Finally, we can obtain several observations (labeled 'x' in Figure 1) with  $T$  discrete time slots since  $T$  intersections are taken into account in the image. After transforming the two-dimensional image into a one-dimensional radial space, we apply a conventional single-target tracking technique since the boundary of any object will be constructed from the obtained trajectory or data association represented by the set of red circles in Figure 1d.

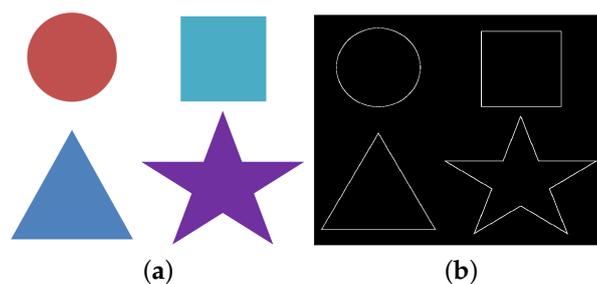
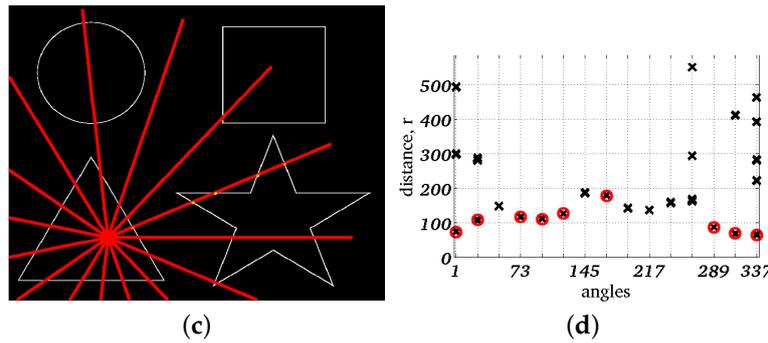


Figure 1. Cont.



**Figure 1.** Motivation and steps of the proposed idea: (a) → (b) → (c) → (d). (a) Raw image; (b) detected points; (c)  $T$  intersections in a radial way; (d) transformed tracking space.

### 3. Mathematical Background

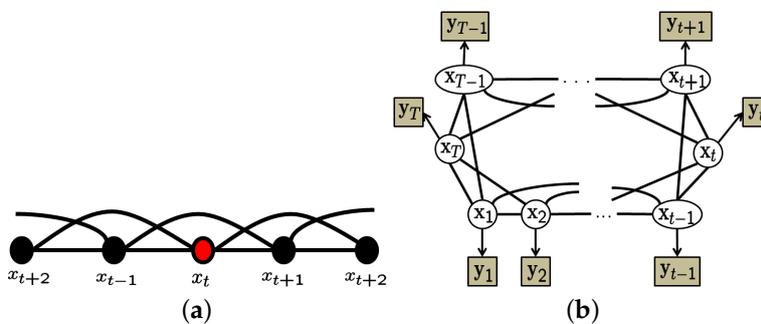
As described in Section 2, boundary tracing is performed with the single-target tracking technique that is popularly used in highly noisy surveillance systems. Since our tracking model is based on an off-line system, we use the idea of Markov chain Monte Carlo data association (MCMCDA) [31] to provide highly accurate data association for edge linking. MCMCDA uses data association to partition the measurements into two sets of data: target-oriented measurements and non-target oriented measurements (clutter). After this data association process, the single-target tracking problem becomes much simpler and straightforward to address [31]. Let the unobserved stochastic process with hidden states  $\{x_t\}$  for  $t \in \{1, 2, \dots, T\}$ ,  $x_t \in \mathbf{X}$ , be modeled as a Gaussian Markov random field (GMRF) process of transition probability  $p(x_t|x_{ne(t)})$  for the system prior, where  $ne(t)$  is a set of the neighbors of the  $t$ -th state.

#### 3.1. Gaussian Markov Random Field

A shape prior is now commonly used for segmenting images in image processing and computer vision [32].

A Markov random field (MRF) is often used in the segmentation of images in image processing and computer vision. For instance, MRF models the relation of neighboring pixels to the coloring foreground and background using a mathematical model, such as the Ising or Potts model [33–35]. There are alternative uses of MRF in computer vision. MRF has been used for the object shape prior, which is used for Bayesian object recognition [36,37]. Our work is also based on the use of a shape prior.

Assume that there is a graphical model that consists of a set of states with the second order Gaussian Markov random field (GMRF) [38] representation in Figure 2a.



**Figure 2.** Used graphical models: (a) the second order Gaussian Markov random field; and (b) circular state space model.

Given the GMRF model, we have an interesting mathematical form of the prior as Equation (1):

$$\begin{aligned}
 p(\mathbf{X}|\kappa) &\propto \kappa^{T/2} \exp \left\{ -0.5\kappa \sum_{t=1}^T (u_0x_t - u_1x_{t-1} - u_1x_{t+1} - u_2x_{t-2} - u_2x_{t+2})' \right. \\
 &\quad \left. \times (u_0x_t - u_1x_{t-1} - u_1x_{t+1} - u_2x_{t-2} - u_2x_{t+2}) \right\} \\
 &= \kappa^{T/2} \exp \left\{ -0.5\kappa (\mathbf{D}\mathbf{X})' (\mathbf{D}\mathbf{X}) \right\} \\
 &= \kappa^{T/2} \exp \left\{ -\frac{1}{2} \mathbf{X}' \mathbf{Q} \mathbf{X} \right\}
 \end{aligned} \tag{1}$$

where  $\mathbf{Q} = \kappa \mathbf{D}' \mathbf{D}$ , and  $\mathbf{D}$  is a circulant matrix defined by:

$$\begin{aligned}
 \mathbf{D} &= \begin{bmatrix} u_0 & -u_1 & -u_2 & 0 & \cdots & 0 & -u_2 & -u_1 \\ -u_1 & u_0 & -u_1 & -u_2 & \cdots & 0 & 0 & -u_2 \\ -u_2 & -u_1 & u_0 & -u_1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ -u_1 & -u_2 & 0 & 0 & \cdots & -u_2 & -u_1 & u_0 \end{bmatrix} \\
 \mathbf{X} &= \begin{bmatrix} x_1, x_2, \dots, x_{T-1}, x_T \end{bmatrix}' .
 \end{aligned}$$

### 3.2. Circular State Space Model

We assume the likelihood for the target-oriented measurements as follows  $y_t = x_t + v_t$ ,  $v_t \sim \mathcal{N}(\cdot; 0, \tau)$ , and the non-target-oriented measurements follow a uniform distribution since there is no prior information. We have the following state and measurement space models in a time series, as shown in Figure 2b. Given the GMRF prior and likelihood function, we have  $\mathbf{y}_{1:T} \sim \mathcal{N}(\cdot; \mathbf{X}_t, \tau \mathbf{I}_{T \times T})$ . We finally have a marginalized likelihood for the circular state space model with observations in the time series by:

$$\begin{aligned}
 p(\mathbf{y}_{1:T}|\tau, \kappa) &= \int p(\mathbf{y}_{1:T}|\mathbf{X}, \theta) p(\mathbf{X}|\theta) d\mathbf{X} \\
 &= \int \mathcal{N}(\mathbf{y}_{1:T}; \mathbf{X}, \tau \mathbf{I}) \mathcal{N}(\mathbf{X}; \mathbf{0}, \mathbf{Q}^{-1}) d\mathbf{X} \\
 &= \mathcal{N}(\mathbf{y}_{1:T}; \mathbf{0}, \tau \mathbf{I} + \mathbf{Q}^{-1}).
 \end{aligned} \tag{2}$$

### 3.3. Data Association for Single-Object Tracking

By assuming that the time series contains a single track, then each observation can be assigned to one of the following cases: the observation is an element of a set of clutter, i.e.,  $\mathbf{w}_0$ , or the observation is a target-oriented signal, i.e.,  $\mathbf{w}_1$ . Therefore, data association partitions the measurements into target-oriented observations ( $\mathbf{w}_1$ ) and clutter ( $\mathbf{w}_0$ ). The motivation of our proposed tracking algorithm is now to search for an appropriate single track  $\mathbf{w}_1$ .

Note that we simplified the notation of the observations by  $\mathbf{y} = \mathbf{y}_{1:T}$  and  $y_{T+i} = y_i$  for  $i > 0$  since it is cyclic. Suppose that  $\mathbf{w}$  is a configuration of  $\mathbf{y}$ , and it is a collection of partitions of  $\mathbf{y}$ . For  $\mathbf{w} \in \Omega$ , we have  $\mathbf{w} = \{\mathbf{w}_0, \mathbf{w}_1\}$ , where  $\mathbf{w}_0 \cup \mathbf{w}_1 = \mathbf{y}$  and  $|\mathbf{w}_0 \cap \mathbf{w}_1| = 0$ ,  $\mathbf{w}_0$  is a set of clutter, false alarms,  $|\mathbf{w}_1 \cap y_t| \leq 1$  for  $t = 1, 2, \dots, T$ , and  $L_{\min} \leq |\mathbf{w}_1| \leq T$ , where  $L_{\min}$  is the minimum length of the proposed track. The cardinality of the set is denoted by  $|\cdot|$ . Let  $\theta = (p_d, \lambda_f)'$  be the set of unknown environmental parameters (EPs), where  $'$  is the transpose operator. In the EPs,  $p_d$  is the detection probability, and  $\lambda_f$  is the false alarm rate per unit time. Our interest is to generate samples from a stationary posterior distribution for the configuration  $\mathbf{w}$ ,

$$p(\mathbf{w}|\mathbf{y}, \tau, \kappa) = \int p(\mathbf{w}|\mathbf{y}, \theta, \tau, \kappa) \pi(\theta|\mathbf{y}, \tau, \kappa) d\theta.$$

The conditional posterior has the form,

$$\begin{aligned} p(\mathbf{w}|\mathbf{y}_{1:T}, \tau, \kappa, \theta) &\propto p(\mathbf{y}_{1:T}|\mathbf{w}, \tau, \kappa, \theta)p(\mathbf{w}|\tau, \kappa, \theta) \\ &= p(\mathbf{y}_{\mathbf{w}_1}|\tau, \kappa)p(\mathbf{y}_{\mathbf{w}_0}|\theta)p(\mathbf{w}|\theta) \end{aligned} \quad (3)$$

where  $p(\mathbf{x}|\cdot)$  follows a Gaussian Markov random field (GMRF) prior.

As mentioned in connection with the original MCMCDA, we can use fixed environmental parameters (EPs),  $\theta$ . When they are known, we can sample only  $\mathbf{w}$  over the conditional posterior distribution,  $p(\mathbf{w}|\mathbf{y}, \theta, \tau, \kappa)$  since the underlying trajectory  $\mathbf{X}$  is marginalized as in Equation (2). Therefore, the stationary posterior distribution of our interest is:

$$\begin{aligned} p(\mathbf{w}|\mathbf{y}_{1:T}, \tau, \kappa, \theta) &= \frac{1}{Z} p(\mathbf{y}_{\mathbf{w}_1}|\tau, \kappa)p(\mathbf{y}_{\mathbf{w}_0})p(\mathbf{w}|\theta) \\ &= \frac{1}{Z} \mathcal{N}(\mathbf{y}_{\mathbf{w}_1}; \mathbf{0}, \tau\mathbf{I} + \mathbf{Q}^{-1}) \prod_{t=1}^T \left(\frac{1}{R}\right)^{f_t} \prod_{t=1}^T Be(d_t; p_d) Po(f_t; \lambda_f R) \end{aligned} \quad (4)$$

where  $Z = p(\mathbf{y}_{1:T}|\theta)$  and  $Be()$ ,  $Po()$  denote the Bernoulli and Poisson distributions, respectively. Here,  $R$  is the maximum distance from the origin for all measured observations, and  $p_d$  and  $\lambda_f$  denote the detection probability and the false alarm occurrence rate  $\lambda_f$ . The observations are divided into two sets:  $\mathbf{y}_{\mathbf{w}_0}$  and  $\mathbf{y}_{\mathbf{w}_1}$ , which are the sets of observations containing the clutter and targets, respectively. In this equation, the parameters, deterministically obtained when the model configuration  $\mathbf{w}$  is proposed, are given by the number of detections at time  $t$ ,  $d_t$ , the number of undetected targets at time  $t$ ,  $u_t = 1 - d_t$ , and the number of false alarms at time  $t$ ,  $f_t = n_t - d_t$ . In addition, we need to consider a special case in which there are no detected observations for certain sequential slots in order to design occlusions for the boundary. For instance, we may not have any observations at times  $a$  and  $b$  in data association; thus, let  $\mathbf{m}$  be a set of the time indexes of the undetected observations. Now, our likelihood can be slightly changed by applying marginalization, and  $\setminus \mathbf{m}$  denotes 'not  $\mathbf{m}$ ', such that:

$$p(\mathbf{y}_{\mathbf{w}_1}|\tau, \kappa) = \int_{\mathbf{m}} \mathcal{N}(\mathbf{y}_{\mathbf{w}_1}, \mathbf{y}_{\mathbf{m}}; \mathbf{0}, \tau\mathbf{I} + \mathbf{Q}^{-1}) d\mathbf{y}_{\mathbf{m}} = \mathcal{N}\left(\mathbf{y}_{\mathbf{w}_1}; \mathbf{0}_{\setminus \mathbf{m}}, \left[\tau\mathbf{I} + \mathbf{Q}^{-1}\right]_{\setminus \mathbf{m}}\right). \quad (5)$$

## 4. Proposed Approach

### 4.1. Improved Posterior with Marginalization of the EPs

The unknown parameters of our interest are both the configuration  $\mathbf{w}$  and the environmental parameters  $\theta$ . Their prior distribution  $p(\mathbf{w}, \theta)$  is decomposed by  $p(\mathbf{w}|\theta)p(\theta)$  where:

$$\begin{aligned} p(\mathbf{w}|\theta) &= \prod_{t=1}^T p_d^{d_t} (1 - p_d)^{1-d_t} \frac{(\lambda_f R)^{f_t} \exp(-\lambda_f R)}{f_t!} \text{ and} \\ p(\theta) &= p(p_d, \lambda_f) = p(p_d)p(\lambda_f) \end{aligned} \quad (6)$$

and  $p_d$  and  $\lambda_f$  are assumed to be independent. Given the conjugate prior distribution of the EPs and configuration, we have the marginalized posterior,

$$\begin{aligned} p(\mathbf{w}|\mathbf{y}_{1:T}, \tau, \kappa) &\propto p(\mathbf{y}_{\mathbf{w}_1}|\tau, \kappa)p(\mathbf{y}_{\mathbf{w}_0}) \int p(\mathbf{w}|\theta)p(\theta) d\theta \\ &= \mathcal{N}\left(\mathbf{y}_{\mathbf{w}_1}; \mathbf{0}_{\setminus \mathbf{m}}, \left[\tau\mathbf{I} + \mathbf{Q}^{-1}\right]_{\setminus \mathbf{m}}\right) \frac{\beta_f^{\alpha_f}}{\prod_{t=1}^T f_t!} \mathcal{B}(\alpha_d, \beta_d) \Gamma(\alpha_f) \end{aligned} \quad (7)$$

where:

$$\alpha_d = \sum_{t=1}^T d_t + 1,$$

$$\begin{aligned}
\beta_d &= T - \sum_{t=1}^T d_t + 1, \\
\alpha_f &= \sum_{t=1}^T f_t + 1 \text{ and} \\
\beta_f &= (TR)^{-1}
\end{aligned} \tag{8}$$

and  $p(\theta)$  is assumed to be a uniform distribution since we do not have any prior information thereof. The detailed derivation is shown in Appendix A.1.

Finally, the log of the marginalized posterior is:

$$\begin{aligned}
\mathcal{L}(\mathbf{w}) = \log p(\mathbf{w}|\mathbf{y}_{1:T}, \tau, \kappa) &= -\frac{1}{2} \log |2\pi\mathbf{S}| - \frac{1}{2} \mathbf{y}_{\mathbf{w}_1}^T \mathbf{S}^{-1} \mathbf{y}_{\mathbf{w}_1} \\
&+ \alpha_f \log \beta_f - \sum_{t=1}^T \sum_{i=1}^{f_t} \log i + \log \mathcal{B}(\alpha_d, \beta_d) + \log \Gamma(\alpha_f)
\end{aligned} \tag{9}$$

where  $\mathbf{S} = [\tau\mathbf{I} + \mathbf{Q}^{-1}]_{\setminus \mathbf{m}}$ .

#### 4.2. Approach Based on the Metropolis–Hastings Algorithm

Markov chain Monte Carlo (MCMC) is one of the most intuitive statistical approaches to constructing a target posterior, and it is applied to image processing [39–41]. We implemented several types of moves for the Metropolis–Hastings (MH) update for linking edges: (a) pop: an observation moves from  $\mathbf{w}_1$  to  $\mathbf{w}_0$ ; that is, an observation that was previously considered as a target-oriented one is now considered in a non-target observation; (b) push: an observation moves from  $\mathbf{w}_0$  to  $\mathbf{w}_1$ ; i.e., an observation that was previously considered as a non-target-oriented observation is re-categorized as a target-oriented observation; and (c) swap: for time  $t$ , if we have a target-oriented observation  $a_t \in \mathbf{w}_1$  and we can choose a non-target-oriented observation  $b_t \in \mathbf{w}_0$  for  $\mathbf{y}_t^{(a_t)}, \mathbf{y}_t^{(b_t)} \in \mathbf{y}_t$  and  $a_t \neq b_t$ , then, we exchange them in the sets. That is, the updated configuration has two updated sets:  $b_t \in \mathbf{w}_1$  and  $a_t \in \mathbf{w}_0$ . We have an acceptance probability for the Metropolis–Hastings algorithm:

$$\mathcal{A} = \min \left\{ 1, \frac{p(\mathbf{w}'|\mathbf{y}, \tau, \kappa)q(\mathbf{w}; \mathbf{w}')}{p(\mathbf{w}|\mathbf{y}, \tau, \kappa)q(\mathbf{w}', \mathbf{w})} \right\}.$$

#### 4.3. Gibbs Sampler-Based Approach

Here, given the mathematical model, we can calculate the posterior distribution of each case of data association at time  $t$ , and a Gibbs sampler can generate the samples from the pre-calculated posterior distribution. From this point of view, we first introduce an auxiliary random variable  $c_t$  that identifies the index of the target-oriented observation at time  $t$  from the data association for  $t \in \{1, 2, \dots, T\}$ , i.e.,  $c_t \in \{0, 1, 2, \dots, N_t\}$ , where  $N_t$  is the number of measurements at time  $t$ . Here,  $c_t = 0$  represents that no target-oriented measurement is detected, and  $c_t = i$  represents that the  $i$ -th observation is detected and is a target-oriented signal at time  $t$ . Now, our interest is to generate samples from the joint full posterior distribution of  $\mathbf{C}_{1:T}$ ,  $\pi(\mathbf{C}_{1:T}|\mathbf{y}, \tau, \kappa)$ , rather than  $p(\mathbf{w}|\mathbf{y}, \tau, \kappa)$ , since the configuration reconstructed by  $\mathbf{C}_{1:t}$  is equivalent to the configuration,  $\mathbf{w}$ , which has been referred to in the previous sections. We draw a sample  $c_t^*$  from the conditional posterior as in Algorithm 1 by  $c_t^* \sim p(c_t^*|\mathbf{c}_{1:t-1}^*, \mathbf{c}_{t+1:T}, \mathbf{y}, \tau, \kappa)$  for time  $t$ . Therefore,

$$p(c_t^*|\mathbf{c}_{1:t-1}^*, \mathbf{c}_{t+1:T}, \mathbf{y}, \tau, \kappa) \propto p(\mathbf{c}_{1:t}^*, \mathbf{c}_{t+1:T}|\mathbf{y}, \tau, \kappa) = p(\mathbf{w}^*|\mathbf{y}, \tau, \kappa) \tag{10}$$

and it is directly estimated by:

$$p(c_t^* = i|\mathbf{c}_{1:t-1}^*, \mathbf{c}_{t+1:T}, \mathbf{y}, \tau, \kappa) = \frac{p(\mathbf{w}^{(t,i)}|\mathbf{y}, \tau, \kappa)}{\sum_{i=0}^{N_t} p(\mathbf{w}^{(t,i)}|\mathbf{y}, \tau, \kappa)} \tag{11}$$

where the configuration  $\mathbf{w}^{(t,i)}$  is constructed with  $\mathbf{c}_{1:T}^{(t,i)} = (\mathbf{c}_{1:t-1}^*, c_t^* = i, \mathbf{c}_{t+1:T})$ .

---

**Algorithm 1** Gibbs sampler-based approach.

---

- 1: Let  $\mathbf{c}_{1:T} = [c_1, c_2, \dots, c_t, \dots, c_T]$  be a vector with non-negative integers s.t.  $c_t \in \{0, 1, \dots, N_t\}$  for all  $t = 1, 2, \dots, T$ .
  - 2: **for**  $iter = 1$  to  $N_{iter}$  **do**
  - 3:     **for**  $t = 1$  to  $T$  **do**
  - 4:         **for**  $i = 0$  to  $N_t$  **do**
  - 5:              $\mathbf{c}_{1:T}^{(t,i)} \leftarrow [\mathbf{c}_{1:t-1}^*, c_t^* = i, \mathbf{c}_{t+1:T}]$  and make  $\mathbf{w}^{(t,i)}$  from the  $\mathbf{c}_{1:T}^{(t,i)}$ .
  - 6:             **end for**
  - 7:             Calculate the posterior  $p(c_t^* = i | \mathbf{c}_{1:t-1}^*, \mathbf{c}_{t+1:T}, \mathbf{y}) = \frac{p(\mathbf{w}^{(t,i)} | \mathbf{y}, \tau, \kappa)}{\sum_{j=0}^{N_t} p(\mathbf{w}^{(t,j)} | \mathbf{y}, \tau, \kappa)}$ .
  - 8:             Draw a sample from the conditional posterior by  $c_t^* \sim p(c_t^* | \mathbf{c}_{1:t-1}^*, \mathbf{c}_{t+1:T}, \mathbf{y}_{1:T})$
  - 9:             **end for**
  - 10:     Reconstruct  $\mathbf{w}$  using  $\mathbf{c}_{1:T}^*$  and find the best configuration  $\hat{\mathbf{w}}$  with the MAP estimate.
  - 11: **end for**
- 

4.4. Variational Bayes Using Mean-Field Approximation with Pseudo-Integration

Variational Bayes (VB) is a particular variational method that aims to find some approximate joint distribution  $q(\mathbf{c}_{1:T}; \theta)$  over the hidden variable  $\mathbf{c}_{1:T}$  to approximate the true joint distribution  $p(\mathbf{c}_{1:T} | \mathbf{y}, \theta)$  by minimizing the Kullback–Leibler (KL) distance (relative entropy),  $KL(q(\mathbf{c}_{1:T}; \theta) || p(\mathbf{c}_{1:T} | \mathbf{y}, \theta))$ . In this case,  $q(\cdot)$  has a simpler form than true distribution  $p(\cdot)$ , and the mean-field form is used to factorize the joint  $q(\mathbf{c}_{1:T})$  into single-variable factors,  $q(\mathbf{c}_{1:T}; \theta) = \prod_t q_t(c_t | \theta)$ . In VB, we wish to obtain a set of distributions  $\{q_t(c_t | \theta)\}_{t=1}^T$  to minimize the KL distance,

$$KL[q(\mathbf{c}_{1:T}; \theta) || p(\mathbf{c}_{1:T} | \mathbf{y}, \theta)] = \int q(\mathbf{c}_{1:T}; \theta) \log \frac{q(\mathbf{c}_{1:T}; \theta)}{p(\mathbf{c}_{1:T} | \mathbf{y}, \theta)} d\mathbf{c}_{1:T}.$$

We can assume that the factorized form via the mean field follows a multinomial distribution, and then, we have:

$$q(\mathbf{c}_{1:T}; \theta) = \prod_{t=1}^T q_t(c_t; \theta) = \prod_{t=1}^T \mathcal{M}(c_t; p_{t,0}, p_{t,1}, \dots, p_{t,N_t}) = \prod_{t=1}^T \prod_{i=0}^{N_t} p_{t,i}^{\delta(c_t=i)}$$

where  $\mathcal{M}(\cdot; \mathbf{p})$  is the multinomial distribution with control parameter  $\mathbf{p}$  and  $\sum_{i=0}^{N_t} p_{t,i} = 1$  for all  $t \in \{1, 2, \dots, T\}$ . An iterative update equation for the factorized single-value form provides:

$$\begin{aligned} \log q(c_t; \theta) &\leftarrow \mathbf{E}_{q(\mathbf{c}_{-t}; \theta)} [\log p(c_t, \mathbf{c}_{-t}, \mathbf{y})] \\ &= \int_{\mathbf{c}_{-t}} q(\mathbf{c}_{-t}; \theta) \log p(c_t, \mathbf{c}_{-t}, \mathbf{y}) d\mathbf{c}_{-t} \\ &= \int_{\mathbf{c}_{-t}} q(\mathbf{c}_{-t}; \theta) [\log p(c_t) + \log p(\mathbf{c}_{-t}) + \log p(\mathbf{y} | c_t, \mathbf{c}_{-t})] d\mathbf{c}_{-t} \\ &= \log p(c_t) + \log \frac{1}{Z} + \int_{\mathbf{c}_{-t}} q(\mathbf{c}_{-t}; \theta) \log p(\mathbf{y} | c_t, \mathbf{c}_{-t}) d\mathbf{c}_{-t}. \end{aligned}$$

Unfortunately, the right-most form of the above equation is not easily estimated, since it does not have a conjugate form; hence, we introduce the second approximated function:

$$q(\mathbf{c}_{-t} | \theta) \approx \tilde{q}(\mathbf{c}_{-t}; \theta) = \begin{cases} 1 & \text{if } \mathbf{c}_{-t} = \bar{\mathbf{c}}_{-t} \\ 0 & \text{otherwise} \end{cases}$$

where  $\bar{c}_{-t}$  are the fixed values that are estimated from the previous step in a way similar to MAP or Gibbs samplers. Given this second approximation, we rewrite:

$$\log q(c_t; \theta) \approx \log \mathcal{M}(c_t; \mathbf{p}_t) + \log p(\mathbf{y}|c_t, \bar{c}_{-t}) + \log \frac{1}{Z}.$$

Here, the second term on the right-hand side is the likelihood we already defined in Equation (4), such that:

$$p(\mathbf{y}|c_t, \bar{c}_{-t}) = p(\mathbf{y}|\mathbf{w}, \theta, \tau, \kappa) = p(\mathbf{y}_{\mathbf{w}_1}|\tau, \kappa)p(\mathbf{y}_{\mathbf{w}_0}) = \mathcal{N}(\mathbf{y}_{\mathbf{w}_1}; \mathbf{0}, \tau\mathbf{I} + \mathbf{Q}^{-1}) \prod_{t=1}^T \left(\frac{1}{R}\right)^{f_t}.$$

Thus, we can calculate the probability mass function for the likelihood directly, and we assign it as:

$$\pi_{t,a} = \frac{p(\mathbf{y}|c_t = a, \bar{c}_{-t})}{\sum_{b \in \{0,1,\dots,N_t\}} p(\mathbf{y}|c_t = b, \bar{c}_{-t})}$$

Then, we write the marginal likelihood as a multinomial distribution by:

$$p(\mathbf{y}|c_t, \bar{c}_{-t}) = \prod_{i=0}^{N_t} \pi_{t,i}^{\delta(c_t=i)}.$$

Therefore, we write:

$$\log q(c_t; \theta) \approx \sum_{i=0}^{N_t} \delta(c_t = i) [\log p_{t,i} + \log \pi_{t,i}] + \log \frac{1}{Z}$$

and:

$$q(c_t; \theta) \approx \frac{1}{Z} \prod_{i=0}^{N_t} (p_{t,i} \pi_{t,i})^{\delta(c_t=i)} = \mathcal{M}(c_t; \mathbf{p}_t^*)$$

where  $p_{t,a}^* = p_{t,a} + \pi_{t,a}$  for all  $t \in \{1, 2, \dots, T\}$  and  $a \in \{0, 1, \dots, N_t\}$ . Here,  $p_{t,a}^*$  is the newly-updated distribution.

The algorithm of the variational approach is described in Algorithm 2.

---

**Algorithm 2** Variational Bayes approach.

---

- 1: Let  $\mathbf{c}_{1:T} = [c_1, c_2, \dots, c_t, \dots, c_T]$  be a vector with non-negative integers s.t.  $c_t \in \{0, 1, 2, \dots, N_t\}$  for all  $t = 1, 2, \dots, T$ .
  - 2: Set  $\mathbf{p}_t = [1/N_t, 1/N_t, \dots, 1/N_t]$  for all  $t \in \{1, 2, \dots, T\}$ .
  - 3: Reconstruct  $\mathbf{w}$  using  $\mathbf{c}_{1:T}$ , and set  $\hat{\mathbf{w}} = \mathbf{w}$ .
  - 4: **while**  $\|\mathbf{c}_{1:T} - \mathbf{c}_{1:T}^*\| > \epsilon$  (convergence) **do**
  - 5:      $\mathbf{c}_{1:T} \leftarrow \mathbf{c}_{1:T}^*$ .
  - 6:     **for**  $t = 1$  to  $T$  **do**
  - 7:         **for**  $i = 0$  to  $N_t$  **do**
  - 8:              $\mathbf{c}_{1:T}^{(t,i)} \leftarrow [\mathbf{c}_{1:i-1}^*, c_i^* = i, \mathbf{c}_{i+1:N}^*]$ , and make  $\mathbf{w}^{(t,i)}$  from  $\mathbf{c}_{1:T}^{(t,i)}$ .
  - 9:             **end for**
  - 10:             Calculate  $\pi_{t,a} = \frac{p(\mathbf{y}|c_t=a, \bar{c}_{-t})}{\sum_{b \in \{0,1,\dots,N_t\}} p(\mathbf{y}|c_t=b, \bar{c}_{-t})}$  for all  $a \in \{0, 1, \dots, N_t\}$ .
  - 11:             Update  $p_{t,a}^* = p_{t,a} + \pi_{t,a}$  for all  $a \in \{0, 1, \dots, N_t\}$ .
  - 12:             Normalize the updated distribution,  $\hat{p}_{t,a}^* = \frac{p_{t,a}^*}{\sum_b p_{t,b}^*}$ .
  - 13:             Obtain  $c_t^* = \arg_a \max \{\hat{p}_{t,a}^*\}_{a=0}^{N_t}$ , and update  $\mathbf{p}_t \leftarrow \hat{\mathbf{p}}_t^*$ .
  - 14:         **end for**
  - 15:     Reconstruct  $\mathbf{w}$  using  $\mathbf{c}_{1:T}^*$ , and find the best configuration  $\hat{\mathbf{w}}$  using the MAP estimate.
  - 16: **end while**
-

#### 4.5. Variance Estimation Using Markov Chain Monte Carlo

The next question is how to estimate  $\kappa$  and  $\tau$ . Unfortunately, neither of these parameters has any closed form, so that we cannot draw the samples using the Gibbs sampler. Alternatively, we can estimate them by using two approaches: (a) the Metropolis–Hastings (MH) algorithm; and (b) prior processing in the training step using fast heuristic algorithms, such as the EM approach. In this paper, we use MCMC to ensure consistency in the Monte Carlo approach since the Gibbs sampler is a type of Monte Carlo.

Samples of the parameters  $(\kappa, \tau)$  are drawn using a standard MH algorithm, such that the unknown parameters are jointly updated according to an acceptance probability:

$$\begin{aligned} \mathcal{A} &= \min \left\{ 1, \frac{p(\tau^*, \kappa^* | \mathbf{y}_{1:T}, \mathbf{w}) q(\tau, \kappa; \tau^*, \kappa^*)}{p(\tau, \kappa | \mathbf{y}_{1:T}, \mathbf{w}) q(\tau^*, \kappa^*; \tau, \kappa)} \right\} \\ &= \min \left\{ 1, \frac{p(\mathbf{y}_{1:T} | \mathbf{w}, \tau^*, \kappa^*) p(\tau^*, \kappa^*) q(\tau, \kappa; \tau^*, \kappa^*)}{p(\mathbf{y}_{1:T} | \mathbf{w}, \tau, \kappa) p(\tau, \kappa) q(\tau^*, \kappa^*; \tau, \kappa)} \right\} \\ &= \min \left\{ 1, \frac{p(\mathbf{y}_{1:T} | \mathbf{w}, \tau^*, \kappa^*)}{p(\mathbf{y}_{1:T} | \mathbf{w}, \tau, \kappa)} \right\} \end{aligned} \quad (12)$$

where  $\tau^*, \kappa^* \sim q(\tau^*, \kappa^*; \tau, \kappa) = p(\tau^*, \kappa^*) = p(\tau^*) p(\kappa^*)$ .

## 5. Results

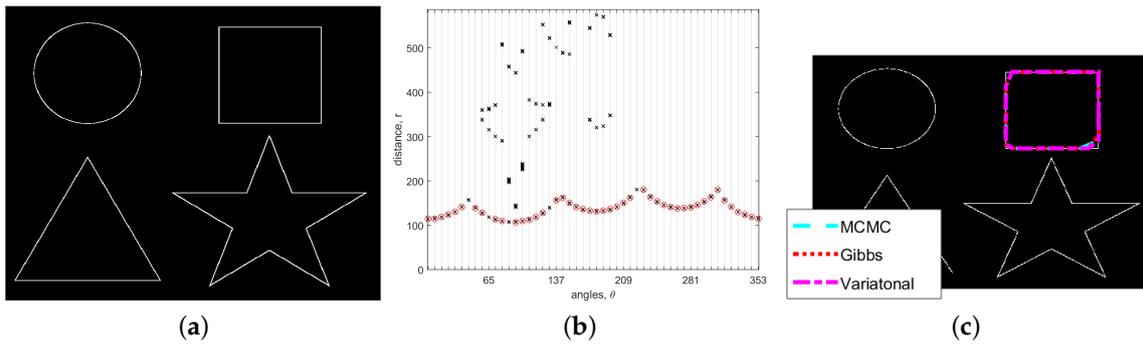
We performed simulations on raw images to evaluate our proposed algorithms. We compared the performance with a popularly-used boundary tracing technique that is a built-in function, “bwtraceboundary”, and our proposed three algorithms. Note that the boundary tracing approach is not used for segmenting objects in an image; instead, it is used for linking points. However, we chose this algorithm for the comparison since our approach mainly involves linking edge points in order to segment the objects. We used the following settings for our proposed algorithms:  $\tau = 0.01$ ,  $\kappa = 0.01$ ,  $T = 50$  and  $N_{iter} = 300$ .

### 5.1. Noise-Free Image

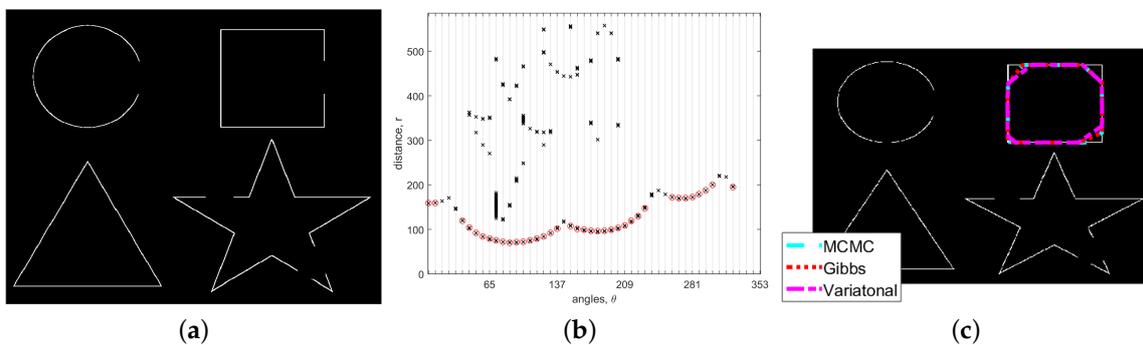
The simulation results we obtained for a noise-free image are as shown in Figure 3. This figure displays three sub-figures: the detected points (left), the trajectory we estimated from the edge points using variational inference (center) and the object segmented using the three proposed approaches (right). Using the interactive segmentation scheme [42,43], we clicked the center of the square in Figure 3a. As we can see in Figure 3b, whereas the detected edge points of uninteresting objects, including “star”, “circle,” and “triangle”, are considered as clutter, the edge points of the square are processed as significant signals with a long trajectory in our proposed approaches. Thus, for a noise-free image, we can obtain proper segmentation using our three proposed approaches as shown in Figure 3c. The elapsed times for this noise-free image are 211.69 (MCMC), 211.67 (Gibbs) and 14.558 (variational Bayes). As expected, the variational Bayes approach is more efficient than the other two approaches for noise-free image segmentation.

### 5.2. Images with Occlusion

Image segmentation is known not to be straightforward for images in which occlusion exists. In order to evaluate the performance of our proposed approaches, we generated a synthetic example with occlusion as shown in Figure 4a. In this figure, the square in the black-and-white image includes occlusion. The red circles in the central sub-figure signify the observations associated with the restored trajectory using variational inference. This figure also shows interesting results in that observations in the corner of the square are not detected in the trajectory, even though they are well detected in the noise-free image in Figure 3a. This is because the observations at the corners are considered to be clutter since the occlusions introduce additional uncertainty in the trajectory.



**Figure 3.** Comparison of segmentation by three different proposed algorithms for a noise-free image. (a) Detected Points; (b) Detected trajectory via variational Bayes; (c) Segmentation.



**Figure 4.** Comparison of the segmentation by the three different proposed algorithms for an image with occlusion. (a) Detected points; (b) Detected trajectory via variational Bayes; (c) Segmentation.

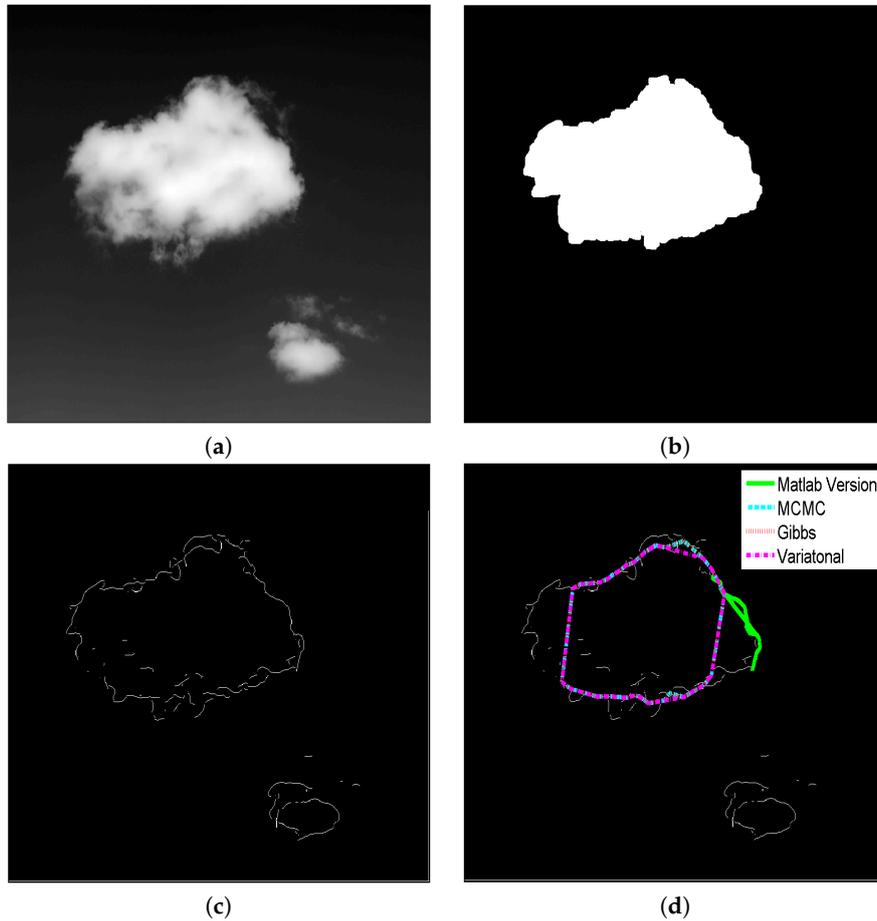
### 5.3. Image Segmentation with Varying Noise Levels

The final aspect we need to investigate as part of our performance evaluation is to check the extent to which the proposed approaches accurately segment objects in an image with varying noise levels. Thus, we added salt and pepper noise at a noise level of  $\sigma_\epsilon$ .

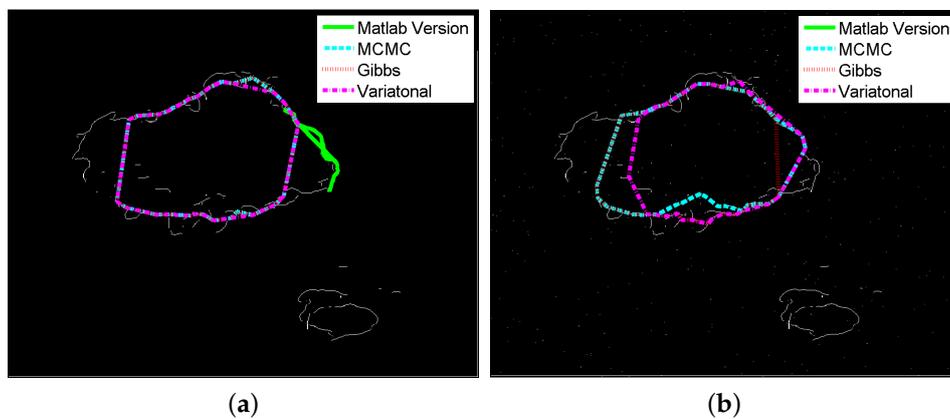
Figure 5 shows (a) the raw image without any added noise  $\sigma_\epsilon = 0$ ; (b) hidden true segmentation; (c) detected points obtained using the “Sobel” detector; and (d) the linked boundary. As can be seen in Figure 5c, many detected points are not connected, although the raw image does not contain any additional noise. We evaluated the performance of our algorithm by comparing two measures: the averaged values and standard deviations of overlapping rates and execution times with 10 runs. The overlapping rate is the similarity between the hidden true segmentation in Figure 5e and the region that is reconstructed from the detected trajectories. Let  $\mathcal{K}$  and  $\mathcal{K}_r$  be the hidden true segmentation and the reconstructed polygon with a small number of detected observations. The similarity is calculated by  $S(\mathcal{K}, \mathcal{K}_r) = \frac{\mathcal{K} \cap \mathcal{K}_r}{\mathcal{K} \cup \mathcal{K}_r}$ . Note that  $S(\mathcal{K}, \mathcal{K}_r)$  has a value between zero and one. As the similarity approaches one, the reconstructed polygon increasingly takes on the desired, hidden true segmentation.

The segmentation results obtained using four different algorithms are shown in Figure 6. As shown in this figure, the “bwtraceboundary” in MATLAB performed very poorly in all cases because it cannot overcome noise. However, our proposed approach performs well even in a severely-cluttered environment as in  $\sigma_\epsilon = 0.005$ . Another interesting finding is the ability of the MCMC-based approach to reliably construct the boundary such that the edges seem invariant. However, the Gibbs sampling and variational Bayes (VB) approaches are adversely affected by the noisy data and perform unstable segmentation. Table 1 presents a more quantitative comparison in terms of the overlapping rate and execution time. As can be seen in this table, MCMC outperforms the other approaches in all cases, and it provides stable results even for  $\sigma_\epsilon = 0.01$ . As we expected, the performance of all approaches

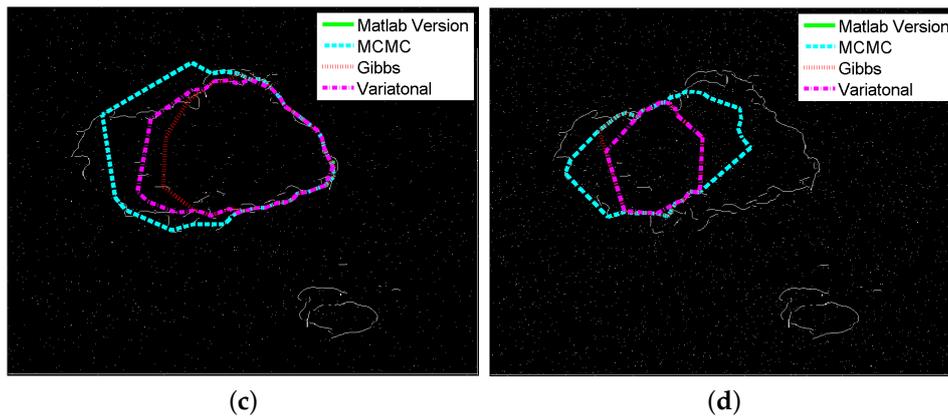
also tends to decrease as more noise is added. Since MCMC and the Gibbs sampler are Monte Carlo simulations, they are rather slower than “bwtraceboundary” and VB. The last important finding is that VB is a practically efficient algorithm for small values of  $\sigma_e$  since it provides a high overlapping rate and its execution time is relatively short compared to that of the Monte Carlo simulations.



**Figure 5.** Raw image, hidden true segmentation for ground truth determination, points detected by the “Sobel” detector and the linked boundaries. (a) Raw image; (b) True segmentation; (c) Detected points; (d) Linked edges.



**Figure 6.** Cont.



**Figure 6.** Comparison of edge linking with varying noisy rate,  $\sigma_\epsilon$ . (a)  $\sigma_\epsilon = 0$ ; (b)  $\sigma_\epsilon = 0.001$ ; (c)  $\sigma_\epsilon = 0.005$ ; (d)  $\sigma_\epsilon = 0.01$ .

**Table 1.** Comparison of the overlapping rate,  $S(\mathcal{K}, \mathcal{K}_r)$  and execution time. VB, variational Bayes.

| Overlapping Rate (Average $\pm$ Std) |                                |                                   |                    |                  |
|--------------------------------------|--------------------------------|-----------------------------------|--------------------|------------------|
| $\sigma_\epsilon$                    | MATLAB                         | MCMC                              | Gibbs              | VB               |
| 0                                    | $0.5 \times 10^{-2} \pm 0.004$ | <b><math>0.77 \pm 0.10</math></b> | $0.71 \pm 0.13$    | $0.71 \pm 0.14$  |
| 0.001                                | $0.2 \times 10^{-2} \pm 0.003$ | <b><math>0.76 \pm 0.08</math></b> | $0.71 \pm 0.09$    | $0.74 \pm 0.08$  |
| 0.005                                | $0.8 \times 10^{-3} \pm 0.003$ | <b><math>0.72 \pm 0.15</math></b> | $0.67 \pm 0.10$    | $0.67 \pm 0.12$  |
| 0.01                                 | $1.2 \times 10^{-3} \pm 0.003$ | <b><math>0.64 \pm 0.14</math></b> | $0.56 \pm 0.13$    | $0.48 \pm 0.14$  |
| Execution Time (Average $\pm$ Std)   |                                |                                   |                    |                  |
| $\sigma_\epsilon$                    | MATLAB                         | MCMC                              | Gibbs              | VB               |
| 0                                    | $0.92 \pm 0.18$                | $196.97 \pm 18.15$                | $215.01 \pm 14.46$ | $13.72 \pm 5.77$ |
| 0.001                                | $0.95 \pm 0.19$                | $236.11 \pm 33.91$                | $238.97 \pm 39.36$ | $13.53 \pm 3.30$ |
| 0.005                                | $1.49 \pm 0.47$                | $272.71 \pm 33.26$                | $294.75 \pm 35.81$ | $14.06 \pm 3.48$ |
| 0.01                                 | $1.97 \pm 0.82$                | $332.64 \pm 73.67$                | $368.88 \pm 72.96$ | $15.20 \pm 4.99$ |

#### 5.4. Application to More Complicated Raw Images

We also tested our proposed approaches on several natural images that were obtained from a few image databases [29,44]. Figure 7 shows the segmentation results processed by the “bwtraceboundary” function and our proposed three approaches. We obtained the edge points from the raw image by using Gaussian convolution and a Canny detection [6]. The first, second, third and fourth rows represent the natural images, edge points obtained by the Canny detection, trajectories associated by variational inference and the final segmentation using the four different approaches.

Figure 7 shows that our proposed approaches outperform “bwtraceboundary” in the segmentation procedure. In addition, the segmentation results of the MCMC simulation are more rigid than those obtained using the Gibbs and variational approaches. We also compared the elapsed times for all approaches. Table 2 indicates that  $\mathcal{T}_{bwtraceboundary}(Matlab) < \mathcal{T}_{VariationalBayes}(VB) \ll \mathcal{T}_{MCMC} \approx \mathcal{T}_{Gibbs}$ .

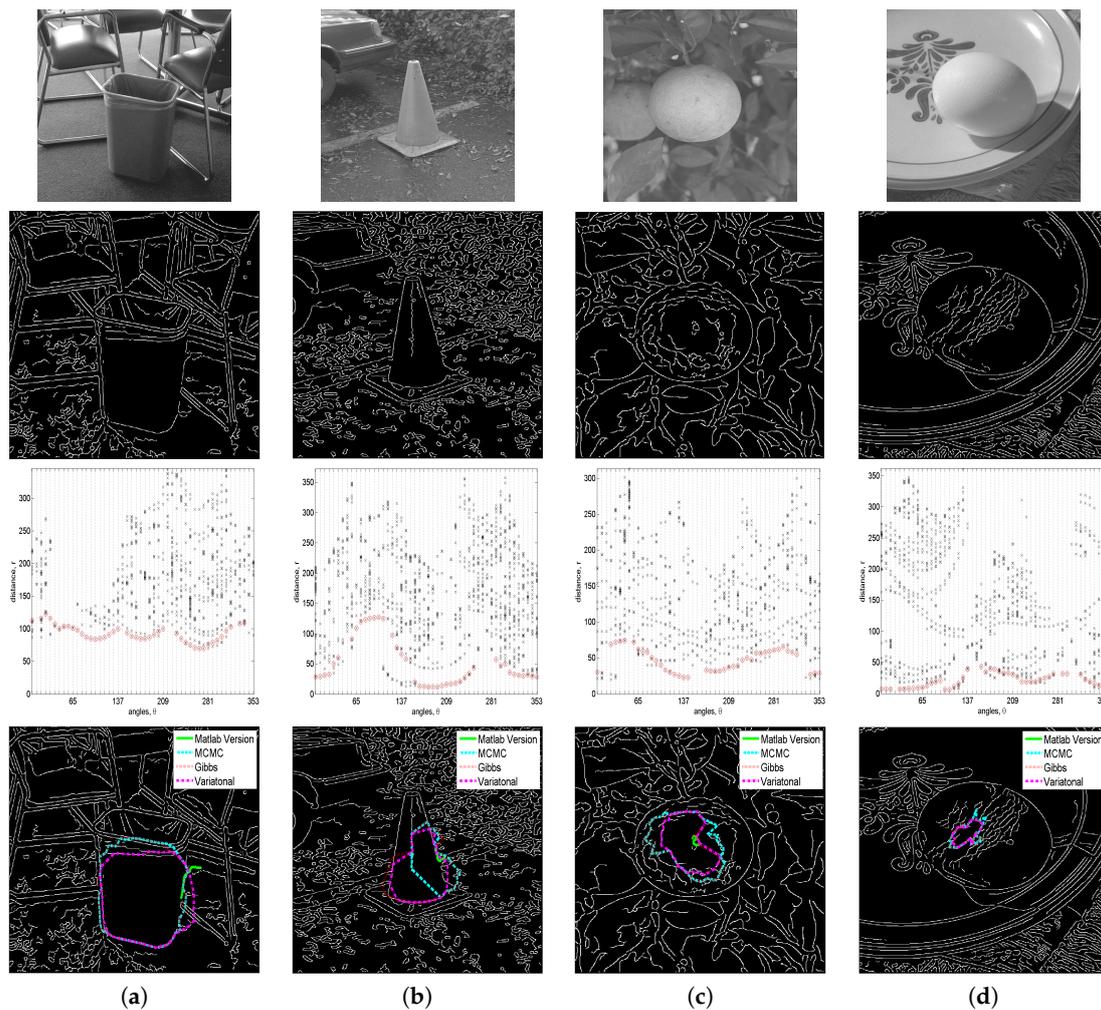


Figure 7. Interactive segmentation via Canny detection.

Table 2. Comparison of the execution times for interactive segmentation via the Canny detection.

| Image | Execution Time |        |        |        |
|-------|----------------|--------|--------|--------|
|       | MATLAB         | MCMC   | Gibbs  | VB     |
| (a)   | 1.7519         | 405.45 | 371.02 | 13.751 |
| (b)   | 2.1913         | 447.51 | 525.08 | 19.322 |
| (c)   | 1.5243         | 355.21 | 434.16 | 9.8595 |
| (d)   | 1.7146         | 359.09 | 450.6  | 10.566 |

We also evaluated the performance of our proposed approaches by changing the detection algorithms. Figure 8 shows the segmentation results for the detected points, which were obtained by the Canny, Bergholm and inversion detection algorithms. It is obvious that our proposed approaches depend on the performance of the edge detection algorithm in use. For instance, the objects in (b), (c) and (d) are well segmented when the inversion detection algorithm is used.

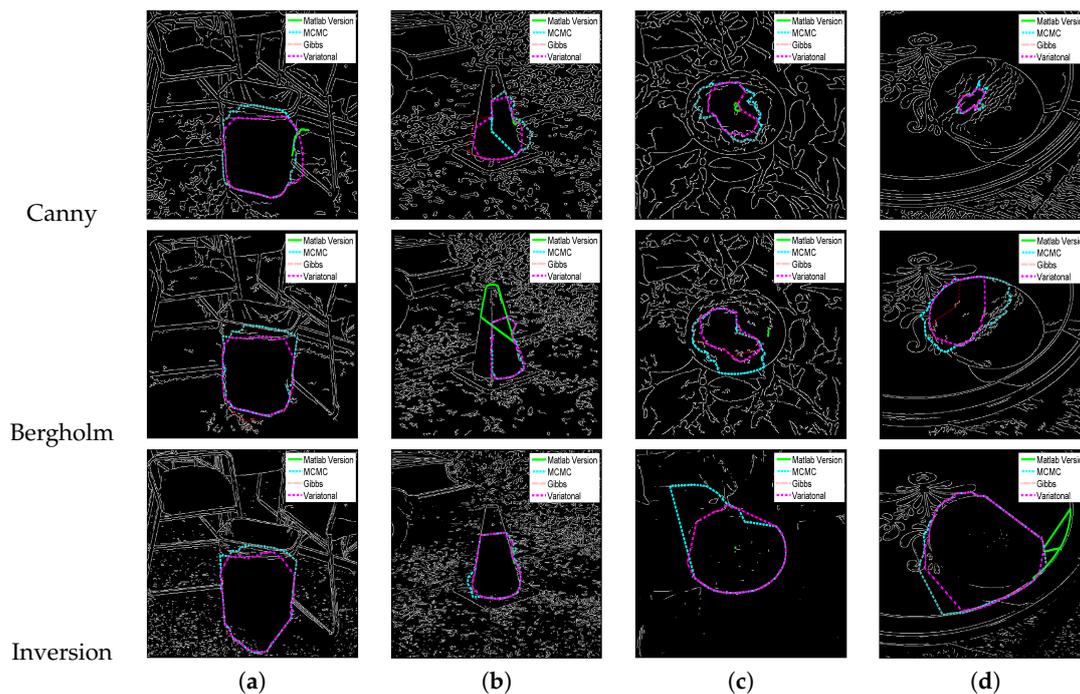


Figure 8. Interactive segmentation via three different detection algorithms.

## 6. Discussion

Apart from the superior performance of our proposed approach in terms of the segmentation of some real images, there are several issues that need to be improved. First, our proposed approach samples the intersecting lines in a radial way such that polygons with a concave form may not be appropriately segmented. Instead of obtaining the concave shape, our proposed approaches are likely to partition the desired concave polygon into several unwanted smaller convex polygons; Second, there are several hyper-parameters that should be carefully selected:  $\kappa$ ,  $\tau$  and  $\mathbf{u}_{0:2} = (u_0, u_1, u_2)$  for the coefficients of the second order GMRF. Unfortunately, these hyper-parameters are very sensitive in the sense that if they are poorly selected, our proposed approaches do not work. Therefore, we need to conduct research on how to automatically select the hyper-parameters used in learning or estimation frameworks; Third, the current approaches do not guarantee stability as they rely on varying origins. The location of the origin determines the velocity and acceleration of the trajectory, which means that  $\kappa$  and  $\tau$  would have to be changed with varying location; Last, we need to improve our proposed approaches to enable them to process multiple objects since the current algorithms only have the ability to accommodate single objects using single-target tracking.

## 7. Conclusions

We introduced a new scheme capable of embedding single-target tracking algorithms, to connect edge points for longer salient edges. Conventional approaches for linking edges are mostly based on the connection of neighboring points rather than global patterns. Therefore, these approaches perform very poorly in the presence of unwanted noise, interruptions and occlusions. However, we can overcome these problems by introducing a probabilistic target-tracking scheme, such as that typically used in cluttered surveillance systems. That is, we converted the edge-linking problem into a target-tracking problem. Given this tracking scheme, we also developed three different algorithms: MCMC, a Gibbs sampler and variational Bayes. MCMC outperforms the other approaches in most cases, although the variational Bayes algorithm seems practically efficient, because it has reasonable accuracy and is computationally effective.

**Acknowledgments:** The author is supported by a new faculty research grant from Korea University.

**Conflicts of Interest:** The author declares no conflict of interest.

## Appendix A

### Appendix A.1. Marginalized Posterior

Given the prior distribution of the EPs and configuration, we can build the marginalized posterior,

$$\begin{aligned}
 p(\mathbf{w}|\mathbf{y}_{1:T}, \tau, \kappa) &\propto p(\mathbf{y}_{\mathbf{w}_1}|\tau, \kappa)p(\mathbf{y}_{\mathbf{w}_0}) \int p(\mathbf{w}|\theta)p(\theta)d\theta \\
 &= \mathcal{N}\left(\mathbf{y}_{\mathbf{w}_1}; \mathbf{0}_{\setminus \mathbf{m}}, \left[\tau\mathbf{I} + \mathbf{Q}^{-1}\right]_{\setminus \mathbf{m}}\right) \\
 &\quad \times \int \prod_{i=1}^T \left(\frac{1}{R}\right)^{f_i} p_d^{d_i}(1-p_d)^{1-d_i} \frac{(\lambda_f R)^{f_i} \exp(-\lambda_f R)}{f_i!} dp_d d\lambda_f \\
 &= \mathcal{N}\left(\mathbf{y}_{\mathbf{w}_1}; \mathbf{0}_{\setminus \mathbf{m}}, \left[\tau\mathbf{I} + \mathbf{Q}^{-1}\right]_{\setminus \mathbf{m}}\right) \\
 &\quad \times \int p_d^{\sum_{i=1}^T d_i} (1-p_d)^{T-\sum_{i=1}^T d_i} \frac{\lambda_f^{\sum_{i=1}^T f_i} \exp(-\lambda_f TR)}{f_i!} dp_d d\lambda_f \\
 &= \mathcal{N}\left(\mathbf{y}_{\mathbf{w}_1}; \mathbf{0}_{\setminus \mathbf{m}}, \left[\tau\mathbf{I} + \mathbf{Q}^{-1}\right]_{\setminus \mathbf{m}}\right) \frac{\beta_f^{\alpha_f}}{\prod_{i=1}^T f_i!} \mathcal{B}(\alpha_d, \beta_d) \Gamma(\alpha_f) \quad (\text{A1})
 \end{aligned}$$

where  $\alpha_d = \sum_{i=1}^T d_i + 1$ ,  $\beta_d = T - \sum_{i=1}^T d_i + 1$ ,  $\alpha_f = \sum_{i=1}^T f_i + 1$ ,  $\beta_f = (TR)^{-1}$  and  $p(\theta)$  is assumed to be a uniform distribution, since we do not have any prior information thereof. Finally, the log of marginalized posterior is  $\mathcal{L}(\mathbf{w}) = \log p(\mathbf{w}|\mathbf{y}_{1:T}, \tau, \kappa) = -\frac{1}{2} \log |2\pi\mathbf{S}| - \frac{1}{2} \mathbf{y}_{\mathbf{w}_1}^T \mathbf{S}^{-1} \mathbf{y}_{\mathbf{w}_1} + \alpha_f \log \beta_f - \sum_{i=1}^T \sum_{j=1}^{f_i} \log j + \log \mathcal{B}(\alpha_d, \beta_d) + \log \Gamma(\alpha_f)$ , where  $\mathbf{S} = [\tau\mathbf{I} + \mathbf{Q}^{-1}]_{\setminus \mathbf{m}}$ .

### Appendix A.2. MATLAB Source Code

A MATLAB source code of the proposed algorithms can be downloaded from the website [45].

## References

1. Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*, 2nd ed.; Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 2001.
2. Lebrun, M.; Buades, A.; Morel, J. A Nonlocal Bayesian Image Denoising Algorithm. *SIAM J. Imaging Sci.* **2013**, *6*, 1665–1688.
3. Yoon, J.W. Statistical denoising scheme for single molecule fluorescence microscopic images. *Biomed. Signal Proc. Control* **2014**, *10*, 11–20.
4. Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66.
5. Marr, D.; Hildreth, E. Theory of Edge Detection. *Proc. R. Soc. Lond. Ser. B Biol. Sci.* **1980**, *207*, 187–217.
6. Canny, J. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *PAMI-8*, 679–698.
7. Meyer, F.; Beucher, S. Morphological segmentation. *J. Vis. Commun. Image Represent.* **1990**, *1*, 21–46.
8. Shih, F.Y.; Cheng, S. Adaptive mathematical morphology for edge linking. *Inf. Sci.* **2004**, *167*, 9–21.
9. Cook, G.W.; Delp, E.J. Multiresolution sequential edge linking. In Proceedings of the 1995 International Conference on Image Processing (ICIP), Washington, DC, USA, 23–26 October; IEEE: New York, NY, USA, 1995; pp. 41–44.
10. Casadei, S.; Mitter, S.K. A hierarchical approach to high resolution edge contour reconstruction. In Proceedings of the 1996 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '96), San Francisco, CA, USA, 18–20 June 1998; IEEE: New York, NY, USA, 1998; pp. 149–154.

11. Herout, A.; Dubska, M.; Havel, J. Review of Hough Transform for Line Detection. In *Real-Time Detection of Lines and Grids*; SpringerBriefs in Computer Science; Springer: London, UK, 2013; pp. 3–16.
12. Farag, A.A.; Delp, E.J. Edge linking by sequential search. *Pattern Recognit.* **1995**, *28*, 611–633.
13. Hancock, E.R.; Kittler, J. Edge-labeling using dictionary-based relaxation. *IEEE Trans. Pattern Anal. Mach. Intell.* **1990**, *12*, 165–181.
14. Christmas, W.J.; Kittler, J.; Petrou, M. Structural matching in computer vision using probabilistic relaxation. *IEEE Trans. Pattern Anal. Mach. Intell.* **1995**, *17*, 749–764.
15. Feng, J.; Shi, Y.; Zhou, J. Robust and Efficient Algorithms for Separating Latent Overlapped Fingerprints. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 1498–1510.
16. Mirmehdi, M.; Petrou, M. Segmentation of color textures. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 142–159.
17. Akinlar, C.; Chome, E. PEL: A Predictive Edge Linking algorithm. *J. Vis. Commun. Image Represent.* **2016**, *36*, 159–171.
18. Yang, X.; Liu, H.; Latecki, L.J. Contour-based object detection as dominant set computation. *Pattern Recognit.* **2012**, *45*, 1927–1936.
19. Jevtic, A.; Melgar, I.; Andina, D. Ant based edge linking algorithm. In *Industrial Electronics, 35th Annual Conference of IEEE Industrial Electronics*; IEEE: New York, NY, USA, 2009; pp. 3353–3358.
20. Wang, Z.; Zhang, H. Edge linking using geodesic distance and neighborhood information. In Proceedings of the IEEE/ASME International Conference on, Advanced Intelligent Mechatronics (AIM 2008), Xi'an, China, 2–5 July 2008; IEEE: New York, NY, USA, 2008; pp. 151–155.
21. Ghita, O.; Whelan, P.F. Computational approach for edge linking. *J. Electron. Imaging* **2002**, *11*, 479–485.
22. Isard, M.; Blake, A. *Contour Tracking by Stochastic Propagation of Conditional Density*; Springer: Berlin/Heidelberg, Germany, 1996; Volume 1, pp. 343–356.
23. Isard, M.; Blake, A. A mixed-state Condensation tracker with automatic model-switching. In Proceedings of the Sixth International Conference on Computer Vision, Bombay, India, 7 January 1998; IEEE: New York, NY, USA, 1998; pp. 107–112.
24. Isard, M.; Blake, A. A smoothing filter for Condensation. In Proceedings of the 5th European Conference Computer Vision, Freiburg, Germany, 2–6 June 1998; Springer: Berlin/Heidelberg, Germany, 1998; Volume 1, pp. 767–781.
25. Zhang, S.; Zhou, H.; Jiang, F.; Li, X. Robust Visual Tracking Using Structurally Random Projection and Weighted Least Squares. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 1749–1760.
26. Zhou, H.; Taj, M.; Cavallaro, A. Target Detection and Tracking With Heterogeneous Sensors. *IEEE J. Sel. Top. Signal Proc.* **2008**, *2*, 503–513.
27. Lim, J.; Zitnick, C.L.; Dollár, P. Sketch Tokens: A Learned Mid-level Representation for Contour and Object Detection. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013.
28. Marimont, D.; Rubner, Y. A probabilistic framework for edge detection and scale selection. In Proceedings of the Sixth International Conference on Computer Vision, Bombay, India, 7 January 1998; IEEE: New York, NY, USA, 1998; pp. 207–214.
29. Gulshan, V.; Rother, C.; Criminisi, A.; Blake, A.; Zisserman, A. Geodesic Star Convexity for Interactive Image Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010.
30. Veksler, O. Star shape prior for graph-cut image segmentation. In Proceedings of the 10th European Conference on Computer Vision (ECCV), Marseille, France, 12–18 October 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 454–467.
31. Oh, S.; Russell, S.J.; Sastry, S. Markov Chain Monte Carlo Data Association for Multi-Target Tracking. *IEEE Trans. Autom. Control* **2009**, *54*, 481–497.
32. Chen, S.; Cremers, D.; Radke, R.J. Image Segmentation with One Shape Prior - A Template-based Formulation. *Image Vis. Comput.* **2012**, *30*, 1032–1042.
33. Barker, S.; Rayner, P. Unsupervised image segmentation using Markov random field models. *Pattern Recognit.* **2000**, *33*, 587–602.
34. Celeux, G.; Forbes, F.; Peyrard, N. *EM-Based Image Segmentation Using Potts Models with External Field*; Rapport de Recherche RR-4456; INRIA: Le Chesnay, France, 2002.

35. Brault, P.; Mohammad-Djafari, A. Unsupervised Bayesian wavelet domain segmentation using Potts-Markov random field modeling. *J. Electron. Imaging* **2005**, *14*, 043011.
36. Rue, H.; Syversveen, A.R. Bayesian object recognition with Baddeley's delta loss. *Adv. Appl. Probab.* **1988**, *30*, 64–88.
37. Chen, S.; Radke, R. Level set segmentation with both shape and intensity priors. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 27 September–4 October 2009; IEEE: New York, NY, USA, 2009; pp. 763–770.
38. Rue, H.; Held, L. *Gaussian Markov Random Fields: Theory and Applications*; Monographs on Statistics and Applied Probability; Chapman & Hall: London, UK, 2005; Volume 104.
39. Fan, A.; Fisher, J., III; Wells, W., III; LEvitt, J.; Willsky, A. MCMC Curve Sampling for Image Segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 477–485.
40. Chang, J.; Fisher, J.W., III. Efficient MCMC Sampling with Implicit Shape Representations. In Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, 20–25 June 2011.
41. Zhou, H.; Li, X.; Sadka, A.H. Nonrigid Structure-From-Motion From 2-D Images Using Markov Chain Monte Carlo. *IEEE Trans. Multimed.* **2012**, *14*, 168–177.
42. Panagiotakis, C.; Papadakis, H.; Grinias, E.; Komodakis, N.; Fragopoulou, P.; Tziritas, G. Interactive image segmentation based on synthetic graph coordinates. *Pattern Recognit.* **2013**, *46*, 2940–2952.
43. Cortés, X.; Serratos, F. An interactive method for the image alignment problem based on partially supervised correspondence. *Expert Syst. Appl.* **2015**, *42*, 179–192.
44. Martin, D.; Fowlkes, C.; Tal, D.; Malik, J. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In Proceedings of the 8th International Conference on Computer Vision, Dallas, TX, USA, 7–14 July 2001; IEEE: New York, NY, USA, 2001; Volume 2, pp. 416–423.
45. Matlab code for Bayesian Edge-Linking Algorithms Using Single-Target Tracking Techniques. Available online: <https://drive.google.com/file/d/0BwXUIJkl6zjeMGN3LU1RVzdiR0E> (accessed on 15 November 2016).



© 2016 by the author; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).