sensors

MDPI

Article

An Interactive Image Segmentation Method in Hand Gesture Recognition

Disi Chen [1], Gongfa Li [1,*], Ying Sun [1], Jianyi Kong [1], Guozhang Jiang [1], Heng Tang [1], Zhaojie Ju [2], Hui Yu [2] and Honghai Liu [2]

[1] School of Machinery and Automation, Wuhan University of Science and Technology, Wuhan 430081, China; chendisi123@126.com (D.C.); sunying6505@hotmail.com (Y.S.); 15697188659@wo.com.cn (J.K.); whjgz@wust.edu.cn (G.J.); cds20161101@163.com (H.T.)

[2] School of Computing, University of Portsmouth, Portsmouth PO1 3HE, UK; zhaojie.ju@port.ac.uk (Z.J.); hui.yu@port.ac.uk (H.Y.); honghai.liu@port.ac.uk (H.L.)

* Correspondence: ligongfa@wust.edu.cn; Tel.: +86-189-0715-9217

Academic Editor: Vittorio M. N. Passaro
Received: 28 October 2016; Accepted: 17 January 2017; Published: 27 January 2017

Abstract: In order to improve the recognition rate of hand gestures a new interactive image segmentation method for hand gesture recognition is presented, and popular methods, e.g., Graph cut, Random walker, Interactive image segmentation using geodesic star convexity, are studied in this article. The Gaussian Mixture Model was employed for image modelling and the iteration of Expectation Maximum algorithm learns the parameters of Gaussian Mixture Model. We apply a Gibbs random field to the image segmentation and minimize the Gibbs Energy using Min-cut theorem to find the optimal segmentation. The segmentation result of our method is tested on an image dataset and compared with other methods by estimating the region accuracy and boundary accuracy. Finally five kinds of hand gestures in different backgrounds are tested on our experimental platform, and the sparse representation algorithm is used, proving that the segmentation of hand gesture images helps to improve the recognition accuracy.

Keywords: image segmentation; Gibbs Energy; min-cut/max-flow algorithm; sparse representation

1. Introduction

Hand gesture recognition, utilized in visual input of controlling computers, is one of the most important aspects in human-computer interaction [1]. Compared with the traditional input methods, such as mice, keyboards and data gloves [2,3], the use of hand gestures to control computers will greatly reduce the user's learning curve and further expand the application scenario. To achieve hand gesture control [4], many research achievements have been conducted by the pioneers in the field. Sophisticated data gloves can capture every single movement of finger joints by highly sensitive sensors [5,6] and store the hand gesture data. The hand gesture recognition process based on computer vision is illustrated in Figure 1. However, some essential problems have yet to be solved. Firstly, the vision-driven hand gesture recognition method is highly dependent on the sensibility of image sensors, therefore the relatively poor image quality hinders its development. Secondly, the image processing algorithms are not robust as they supposed to be, some of which cannot meet the demand to finish the segmentation correctly, while others fulfill the accuracy demands, but require too many human interactions [7], which are not efficient in real applications.

To address the above problems, with the cutting edge technologies, the image sensor industry has mushroomed recently. On the one hand, new kinds of image sensors, like the Microsoft Kinect 2.0, or Asus Xtion, have come into the commercial market [8], and the innovative infrared camera [9] makes it possible obtain depth information from image sensors. On the other hand, innovations in image

processing algorithms have made them capable of segmenting accurate hand gestures, promoting in turn the accuracy of classifiers to ascribe gestures into different patterns.

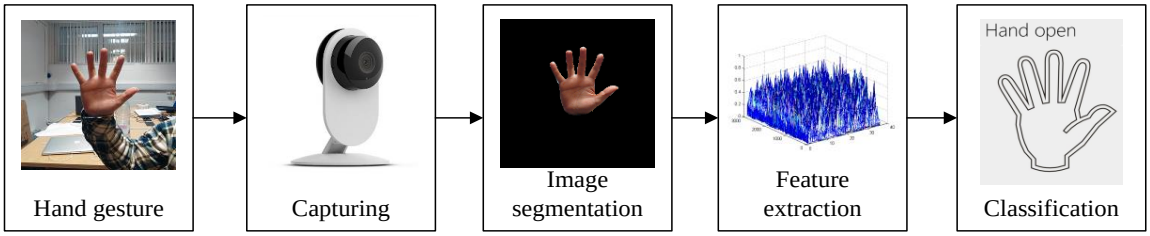


Figure 1. Process of hand gesture recognition.

The image segmentation is an important stage in the whole hand gesture recognition process, and several well-known segmentation methods have been proposed to meet different image segmentation demands. For example, in the graph cut method [10], proposed by Boykov and Jolly, the main idea was to divide one image into "object" and "background". A gray scale histogram was established to describe the distribution of gray scale, and then a cut was drawn to divide the object and background. Max-flow/min cut algorithm was applied to minimize the energy function of one cut, and the segmentation was achieved by this minimized cut. These algorithms not only focus on the whole image, but also take every morphological detail into account. Random walker [11,12] is another supervised image segmentation method, where the image is viewed as an electric circuit. The edges are replaced by passive linear resistors, and the weight of each edge equals the electrical conductance. It proved to perform better segmentation compared with the graph cut method. Gulshan et al. [13] proposed an interactive image segmentation method, which regarded shape as a powerful cue for object recognition, making the problem well posed. The use of geodesic-star convexity made it have a much lower error rate compared with Euclidean star-convexity.

In the process of hand gesture recognition [13], the feature extraction is also very important. The image feature methods such as HOG [14], Hu invariant [15] and Haar [16] are used. In this paper, as for classifier and template matching algorithms, the sparse representation will be applied, since it requires much less sample for training. With the intention of recognising five different hand gestures, according to the dataset of hand gesture images, a dictionary will also be built. Then the K-SVD [17] algorithm is adapted for sample training, and the algorithm will be evaluate and compared with other methods.

2. Modelling of Hand Gesture Images

In order to optimize the segmentation, the human visual system was carefully studied. Our eyes usually got a fuzzy picture of the whole scene at first, and then the saccadic eye movements [18] help us to obtain the details of regions of interest. With the inspiration of the human visual system, we used the Gaussian Mixture Model (GMM) [19] to get an overall view the color distributions of the image. Since the color images are mainly represented in digital formats, with tens of thousands of pixels in one image made up of red, green and blue sub-pixels, as shown in Figure 2, an $M \times N \times 3$ array was applied to store the color information in one image, where M is the horizontal resolution and N is the vertical.

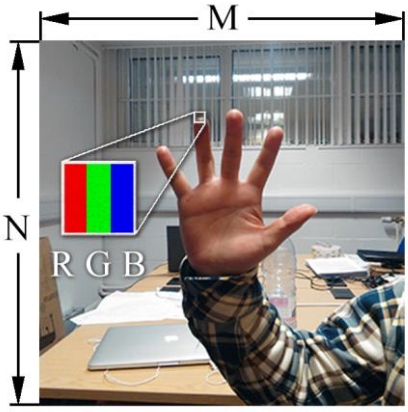


Figure 2. The RGB format hand gesture image.

2.1. Single Gaussian Model

The single Gaussian distribution, also known as the normal distribution [20], was proposed by the French scientist Moivre in 1733. The probability density function of a single Gaussian distribution is given by the formula:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{1}$$

where μ is the mathematical expectation or the mean, σ is the covariance of Gaussian distribution, and exp denotes the exponential function. For convenience, the single Gaussian distribution is usually denoted as:

$$X \sim N(\mu, \sigma^2) \tag{2}$$

The single Gaussian distribution formula is capable of dealing with gray scale pictures, because the variable x has only one dimension. One color image is an M × N × 3 array, so any element $\boldsymbol{x}_i$ in dataset $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ should be at least 3-dimensional. To address this problem, the concept of the multi-dimensional Gaussian distribution is introduced. The definition of d dimensional Gaussian distribution is:

$$N(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left[-\frac{(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu})}{2}\right] \tag{3}$$

where $\boldsymbol{\mu}$ is a d dimensional vector, and as for the RGB model, each component of $\boldsymbol{\mu}$ represents the average red, green and blue color density value. Σ is the covariance matrix and Σ^{-1} is its inverse matrix. $(\boldsymbol{x}-\boldsymbol{\mu})^T$ is the transposed matrix of $(\boldsymbol{x}-\boldsymbol{\mu})$. To simplify Equation (3) above, θ is introduced to represent the parameters $\boldsymbol{\mu}$ and Σ, then the probability density function of the d dimensional Gaussian distribution can be written as:

$$p(\boldsymbol{x}) = N(\boldsymbol{x}; \theta). \tag{4}$$

According to the law of large numbers, every pixel is one sample of the real scene. When the resolution is high enough, the average color density could be estimated.

2.2. Gaussian Mixture Model of RGB Image

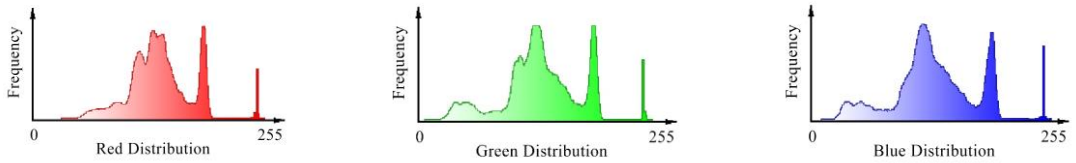
In reality, the color distributions of the gesture image in Figure 2 can be represented by three histograms [21], shown in Figure 3. With independent red, green and blue distributions shown in Figure 3, we can notice that the gesture image cannot be exactly described by one single Gaussian model. But there are about five peaks in each histogram, so five single Gaussian models should be applied in gesture image modelling.

GMM is introduced to approximate the continuous probability distribution by increasing the number of single Gaussian models. The probability density function of GMM with k mixed Gaussian models becomes:

$$p(\boldsymbol{x}) = \sum_{i=1}^{k} \pi_i p_i(\boldsymbol{x}; \theta_i) \tag{5}$$

$$p(\boldsymbol{x}) = \sum_{i=1}^{k} \pi_i N_i(\boldsymbol{x}; \mu_i, \Sigma_i), \tag{6}$$

where $i \in \{1, 2, \ldots, k\}$ shows which single Gaussian model the component belongs to. π_i is the mixing coefficients of k mixed component [22] or the prior probability of $\boldsymbol{x}$ belonging to the i-th single Gaussian model, and $\sum_{i=1}^{k} \pi_i = 1$. $p_i(\boldsymbol{x}; \theta_i)$ is the probability density function of the i-th single Gaussian model, parameterized by μ_i and Σ_i in $N_i(\boldsymbol{x}; \boldsymbol{\mu}_i, \Sigma_i)$. Θ is introduced as a parameters [23] set, $\{\pi_1, \pi_2, \ldots, \pi_k, \theta_1, \theta_2, \ldots, \theta_k\}$, to denote α_i and θ_i.



(**a**) (**b**) (**c**)

Figure 3. Color distributions of the gesture image. (**a**) Red distribution; (**b**) green distribution; (**c**) blue distribution.

As mentioned above, one RGB hand gesture image could be described in the dataset $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, and if we regard $\boldsymbol{X}$ as a sample, its probability density is:

$$p(\boldsymbol{X}; \Theta) = \prod_{j=1}^{n} p(\boldsymbol{x}_j; \Theta) = L(\Theta; \boldsymbol{X}), x_j \in \boldsymbol{X}, \tag{7}$$

where $L(\boldsymbol{X}; \Theta)$ is called likelihood function of parameters given the sample $\boldsymbol{X}$. Then we hope to find a set of parameter Θ to finish modelling. According to maximum likelihood method [24], our next task is to find $\hat{\Theta}$ where:

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}}\, L(\Theta; \boldsymbol{X}). \tag{8}$$

The function $L(\Theta; \boldsymbol{X})$ and $L(\boldsymbol{X}; \Theta)$ have the same equation form, but considering now we are going to use $\boldsymbol{X}$ to estimate Θ, the Θ becomes variables and $\boldsymbol{X}$ are the fixed parameters, it is denoted in the second form. The value of $p(\boldsymbol{X}; \Theta)$ is usually too small to be calculated by computer, so we are going to replace it with the log-likelihood function [25]:

$$\ln(L(\Theta; \boldsymbol{X})) = \ln\left[\prod_{j=1}^{n} p(\boldsymbol{x}_j; \Theta)\right] \tag{9}$$

$$= \sum_{j=1}^{n} \ln\left[\sum_{i=1}^{k} \pi_i p_i(\boldsymbol{x}_j; \theta_i)\right]. \tag{10}$$

2.3. Expectation Maximum Algorithm

After establishing the Gaussian mixture model of a RGB hand gesture image, there are still several parameters that need to be estimated. The expectation maximum (EM) algorithm [26] is introduced for the subsequent calculations. The EM algorithm is a method of acquiring the parameters set Θ in

the maximum likelihood method. There are two steps in this algorithm, called the E-step and M-step, respectively. To start the E-step we will introduce another probability $Q_i(\boldsymbol{x}_j)$. It is a posterior probability of π_i, in another words, the posterior probability of each $\boldsymbol{x}_j$ belonging to the i-th single Gaussian model, from the dataset $\boldsymbol{X}$.

$$Q_i(x_j) = \frac{\pi_i p_i(\boldsymbol{x}_j; \theta_i)}{\sum_{t=1}^{k} \pi_t p_t(\boldsymbol{x}_j; \theta_t)}, \tag{11}$$

where the definition of $Q_i(\boldsymbol{x}_j)$ is given according to Bayes' theorem, and $\sum_{i=1}^{k} Q_i(\boldsymbol{x}_j) = 1$. Then we use Equation (11) to modify the log-likelihood function in (10):

$$\ln(L(\Theta; \boldsymbol{X})) = \sum_{j=1}^{n} \ln \left[\sum_{i=1}^{k} Q_i(\boldsymbol{x}_j) \frac{\pi_i p_i(\boldsymbol{x}_j; \theta_i)}{Q_i(\boldsymbol{x}_j)} \right] \tag{12}$$

$$\geq \sum_{j=1}^{n} \sum_{i=1}^{k} Q_i(\boldsymbol{x}_j) \ln \left[\frac{\pi_i p_i(\boldsymbol{x}_j; \theta_i)}{Q_i(\boldsymbol{x}_j)} \right]. \tag{13}$$

From (12) to (13), the Jensen's inequality have been applied, since $ln''(x) = -\frac{1}{x^2} \leq 0$, it is concave on its domain. Then:

$$\ln \left[\sum_{i=1}^{k} Q_i(\boldsymbol{x}_j) \frac{\pi_i p_i(\boldsymbol{x}_j; \theta_i)}{Q_i(\boldsymbol{x}_j)} \right] \geq \sum_{i=1}^{k} Q_i(\boldsymbol{x}_j) \ln \left[\frac{\pi_i p_i(\boldsymbol{x}_j; \theta_i)}{Q_i(\boldsymbol{x}_j)} \right], \tag{14}$$

Maximizing Equation (13) guarantees that $\ln(L(\Theta; \boldsymbol{X}))$ is maximized. The iteration of an EM algorithm estimating the new parameters in terms of the old parameters is given as follows:

- *Initialization*: Initialize $\boldsymbol{\mu}_{i0}$ with random numbers [27], and the unit matrices are used as covariance matrices Σ_{i0} to start the first iteration. The mixed coefficients or prior probability is assumed as $\pi_{i0} = \frac{1}{k}$.
- *E-step*: Compute the posterior probability of π_i using current parameters:

$$Q_i(\boldsymbol{x}_j) := \frac{\pi_i p_i(\boldsymbol{x}_j; \theta_i)}{\sum_{t=1}^{k} \pi_t p_t(\boldsymbol{x}_j; \theta_t)} = \frac{\pi_i N(\boldsymbol{x}_j; \boldsymbol{\mu}_i, \Sigma_i)}{\sum_{t=1}^{k} \pi_t N(\boldsymbol{x}_j; \boldsymbol{\mu}_t, \Sigma_t)} \tag{15}$$

- *M-step*: Renew the parameters:

$$\pi_i := \frac{1}{n} \sum_{j=1}^{n} Q_i(\boldsymbol{x}_j) \tag{16}$$

$$\mu_i := \frac{\sum_{j=1}^{n} Q_i(\boldsymbol{x}_j) \boldsymbol{x}_t}{\sum_{j=1}^{n} Q_i(\boldsymbol{x}_j)} \tag{17}$$

$$\Sigma_i := \frac{\sum_{j=1}^{n} Q_i(\boldsymbol{x}_j)(\boldsymbol{x}_j - \mu_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T}{\sum_{j=1}^{n} Q_i(\boldsymbol{x}_j)} \tag{18}$$

For most hand gesture images, the number of iterations is usually defined as a certain number. In order to improve the segmentation quality and to take account of the efficiency, the number of iterations should be 8 [28].

3. Interactive Image Segmentation

The modelling method discussed previously provides a universal way of dealing with hand gesture images. To segment the digital images, a mask is introduced as shown in Figure 4, which is a binary bitmap denoted as $\boldsymbol{\alpha}$. By introducing it, we changed the segmentation problem into a pixels labelling problem. As $\alpha_j \in \{1,0\}$, the value 0 is taken for labelling background pixels and 1 for foreground pixels.

Original image Mask

Figure 4. The mask.

To deal with the GMM tractably, we introduce two independent k-component GMMs, one for the foreground modelling and one for the background modelling. Each pixel $\boldsymbol{x}_j$, either from the background or the foreground model, is marked as $\alpha_j = 1$ or 0. The parameters of each component become: $\theta_i = \{\pi_i(\alpha_j), \mu_i(\alpha_j), \Sigma_i(\alpha_j); \alpha_j = 0,1, i = 1, \ldots, k\}$.

3.1. Gibbs Random Field

The overall color modelling completes the first step in our human visual system, to take every detail of the image into account, Gibbs random field (GRF) [29] is introduced. GRF is defined as:

$$P(\boldsymbol{A} = \boldsymbol{a}) = \frac{1}{Z(T)} \exp(-\frac{1}{T} E(\boldsymbol{\alpha})), \tag{19}$$

Here, $P(\boldsymbol{A} = \boldsymbol{a})$ gives the probability of the system $\boldsymbol{A}$ being in the state $\boldsymbol{a}$. T is a constant parameter, whose unit is temperature in physics, and usually its value is 1. $Z(T)$ is the partition function, and:

$$Z(T) = \sum_{a \in A} \exp(-\frac{1}{T} E(\boldsymbol{a})), \tag{20}$$

where, $E(\boldsymbol{\alpha})$ is interpreted as the energy function of the state $\boldsymbol{a}$, to apply GRF in image segmentation, the Gibbs Energy [30] can be defined as follows:

$$E(\boldsymbol{\alpha}) = E(\alpha, \Theta, \boldsymbol{X}) = E(\alpha, i, \theta, \boldsymbol{X}) = U(\alpha, i, \theta, \boldsymbol{X}) + V(\alpha, \boldsymbol{X}) \tag{21}$$

The term $U(\alpha, i, \theta, \boldsymbol{X})$, also called regional term, is defined taking account of GMM. It indicates the penalty of x_j being classified in the background or foreground:

$$U(\alpha, i, \theta, \boldsymbol{X}) = \sum_{j=1}^{n} -\ln[p_i(\boldsymbol{x}_j) \times \pi_i(\alpha_j)], \tag{22}$$

$$= \sum_{j=1}^{n} \left\{ -\ln[\pi_i(\alpha_j)] - \ln[\frac{1}{2}\ln|\Sigma_i(\alpha_j)|] + \frac{1}{2}[\boldsymbol{x}_j - \boldsymbol{\mu}_i(\alpha_j)]^T \Sigma_i(\alpha_j)^{-1}[\boldsymbol{x}_j - \boldsymbol{\mu}_i(\alpha_j)] \right\}. \quad (23)$$

and $V(\alpha, \boldsymbol{X})$, which is the boundary term, which is defined to describe the smoothness between pixel $\boldsymbol{x}_u$ and its neighbour pixels $\boldsymbol{x}_v$ in the pixel set $\boldsymbol{N}$:

$$V(\alpha, \boldsymbol{X}) = \gamma \sum_{x_u, x_v \in N} [\alpha_u \neq \alpha_v] \exp(-\beta \|\boldsymbol{x}_u - \boldsymbol{x}_v\|^2), \quad (24)$$

where the constant γ was obtained as 50 by optimizing the efficiency over training. $[\alpha_u \neq \alpha_v]$ is an indicator function taking values 0 or 1, by judging the formula inside. β is a constant, which represents the contrast of the pixel set $\boldsymbol{N}$, to adjust the exponential term. $E(x)$ in the equation below is the expectation:

$$\beta = \frac{1}{2 \underset{x_u, x_v \in N}{E} [(\boldsymbol{x}_u - \boldsymbol{x}_v)^T (\boldsymbol{x}_u - \boldsymbol{x}_v)]} \quad (25)$$

3.2. Automatical Seed Selection

Until now all the constants have been defined. To begin with, all the pixels in the picture are automatically marked as undefined and labeled $\boldsymbol{U}$ [31]. $\boldsymbol{B}$ is the background seed pixel set and $\boldsymbol{O}$ is the foreground seed set. After the training over training set $\boldsymbol{X}$, the set $\boldsymbol{O}$ is obtained as the segmentation result and $\boldsymbol{O} \subset \boldsymbol{U}$. Three pixel sets are shown in Figure 5.

(a) Original image (b) Image with seed points (c) The segmentation result

Figure 5. The relationships between three pixel sets.

To achieve the segmentation automatically, we propose an initial seeds selection method in hand gesture images. Considering that the human skin color has an elliptical distribution in *YCbCr* color space [32], the image is transformed from RGB color space to *YCbCr*, using the equation below:

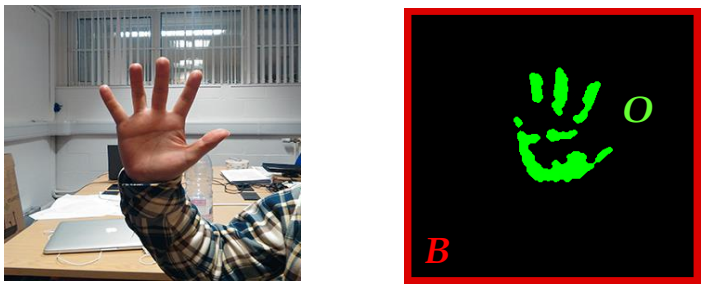
$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \frac{1}{256} \cdot \begin{bmatrix} 65.738 & 129.057 & 25.06 \\ -37.945 & -74.494 & 112.43 \\ 112.439 & -94.154 & -18.28 \end{bmatrix} \cdot \begin{bmatrix} r \\ g \\ b \end{bmatrix}, \quad (26)$$

where, Y indicates the luminance. By setting $Y \in (0, 80)$, the interference of highlights would be overcome. Then the Cb, Cr values of human skin color are located by the elliptical equations given below:

$$\begin{cases} \frac{(x-1.6)^2}{26.39^2} + \frac{(y-2.41)^2}{14.03^2} < 1 \\ \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(2.53) & \sin(2.53) \\ -\sin(2.53) & \cos(2.53) \end{bmatrix} \cdot \begin{bmatrix} Cb - 109.38 \\ Cr - 152.02 \end{bmatrix} \end{cases}, \quad (27)$$

where, x and y are the intermediate variables. All the pixels satisfying the equations above will be marked as the foreground seeds, which belong to set $\boldsymbol{O}$. We also define the pixels on the image edges

as background seeds, which belong to set B, because the gestures are usually located far away from the edges of the images. The result of seeds selection are displayed in Figure 6 below.



Original image Seed map

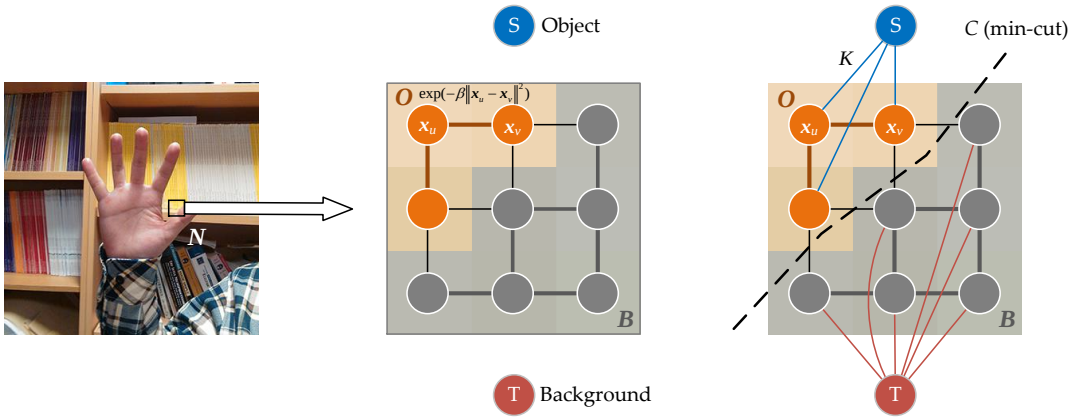
Figure 6. The result of automatic seed selection.

3.3. Min-Cut/Max-Flow Algorithm

According to the Gibbs random field, the image segmentation or pixel labelling problem equals minimizing the Gibbs energy function:

$$\min_{\{\alpha_i; i \in U\}} [\min_{i} E(\alpha, i, \theta, \boldsymbol{X})] \tag{28}$$

The min-cut/max-flow algorithm [33] is proposed to finish the segmentation more accurately. The idea of this algorithm is to regard one image as a net with nodes, and each node take the place of a corresponding pixel. Apart from that, two extra nodes, S and T, are introduced, which represent "source" and "sink", respectively. Node S is linked to pixels belonging to O, while T linked pixels in B as shown in Figure 7.



(a) Original image (b) Nodes and net map (c) Segmentation

Figure 7. Nodes and net model.

There are three kinds of links in the neighbourhood $\boldsymbol{N}$, from pixel to pixel, from pixel to S and from pixel to T, denoted as $\overline{x_u x_v}$, $\overline{x_u \mathrm{S}}$, $\overline{x_u \mathrm{T}}$. Each link is assumed with a certain weight or a cost [34] while it being cut down, which detailed in Table 1.

Table 1. The weight of each link.

Link Type	Weight	Precondition
$\overline{x_u x_v}$	$\exp\left(-\beta\|x_u - x_v\|^2\right)$	$x_u, x_v \in \boldsymbol{N}$
$\overline{x_u \mathrm{S}}$	$U(\alpha = 0, i, \theta, \boldsymbol{X})$	$x_u \in \boldsymbol{U}$
	K	$x_u \in \boldsymbol{O}$
	0	$x_u \in \boldsymbol{B}$
$\overline{x_u \mathrm{T}}$	$U(\alpha = 1, i, \theta, \boldsymbol{X})$	$x_u \in \boldsymbol{U}$
	0	$x_u \in \boldsymbol{O}$
	K	$x_u \in \boldsymbol{B}$
where $K = 1 + \max_{x_u \in \boldsymbol{X}} \sum_{x_u, x_v \in \boldsymbol{N}} \exp(-\beta\|x_u - x_v\|^2)$		

According to the max-flow/min-cut theorem, an optimal segmentation is defined by the minimum cut C as seen in Figure 7c. C is known as a set of $\overline{x_u x_v}$ links, so that:

$$|C| = \sum_{x \in \boldsymbol{U}} U(C, i, \theta, \boldsymbol{x}) + \sum_{x \in \boldsymbol{N}} V(C, \boldsymbol{x}) \tag{29}$$

$$= E(C, i, \theta, \boldsymbol{X}) - \left[\sum_{x \in \boldsymbol{O}} U(\alpha = 1, i, \theta, \boldsymbol{x}) + \sum_{x \in \boldsymbol{B}} U(\alpha = 0, i, \theta, \boldsymbol{x})\right] \tag{30}$$

Then the Gibbs energy could be minimized by using the min-cut defined above. The whole process of this segmentation is as follows: firstly, assign the GMM components i to each $x_j \in \boldsymbol{U}$ according to the human select of the $\boldsymbol{U}$ region. Secondly, the parameters set Θ is learned from the whole pixel set $\boldsymbol{X}$. Thirdly, use the min-cut to minimize the Gibbs energy of the whole image. Then jump to the first step to start another round, and after eight times, the optimal segmentation will be achieved.

4. Experimental Comparison

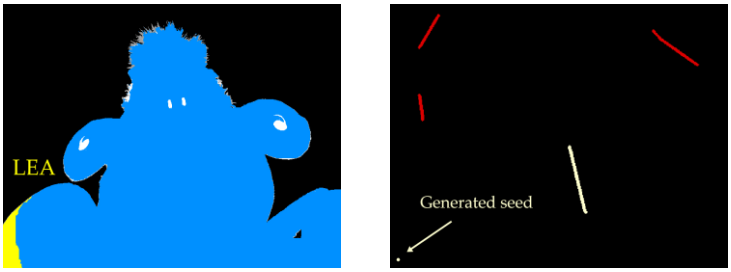
To evaluate interactive segmentation quantitatively, an image dataset proposed by Gulshan [13], which contains 49 images from GrabCut dataset [35], 99 images from PASCAL VOC'09 segmentation challenge [36] and 3 images from the alpha-matting dataset [37] is chosen. Those images cover all kinds of shapes, textures and backgrounds. The corresponding ground true images together with the initial seeds were also included in this dataset. The initial seed maps were made up of 4 manually generated brush-strokes all in 8 pixels wide, and one for foreground and 3 for background as shown in Figure 8.

One image from dataset Ground true image Initial seeds

Figure 8. The evaluation samples from dataset.

To simulate the human interactions, after the first segmentation with initial seed map, one more seed would be generated in the largest connected segmentation error area (LEA) automatically.

As shown in Figure 9a, the blue area is the segmentation result of the algorithm, while the white one is the ground true segmentation and the LEA is marked in yellow. From Figure 9b, the seed is a round dot (8 pixels in diameter), generated according to the LEA. Then we update the segmentation with all the seeds. After that, this step is repeated 20 times, and a sequence of segmentations will be obtained.



(a) The largest connected segmentation error area

(b) One generated seed on seed map

Figure 9. Evaluation on the dataset.

To evaluate the quality of segmentation results, we used two different methods in evaluating the region accuracy (*RA*) and boundary accuracy (*BA*). Each evaluation will be conducted to a single segmentation, and all the images in Gushan's dataset will be tested to verify that our proposed method is suitable for interactive image segmentation.

4.1. Region Accuracy

The *RA* of segmentation results is evaluated by a weighted $F_\beta - measure$ [38]. Compared with normal $F_\beta - measure$, the two terms *Precision* and *Recall* become:

$$Precision^w = \frac{TP^w}{TP^w + FP^w} \tag{31}$$

$$Recall^w = \frac{TP^w}{TP^w + NP^w} \tag{32}$$

where, *TP* denotes the overlap of ground true and segmented foreground pixels. *FP* is the wrongly segmented pixels compared with ground true images and *NP* represent the wrongly segmented background pixels.

The $F_\beta^w - measure$ is defined as follows:

$$RA = F_\beta^w = (1 + \beta^2)\frac{Precision^w \cdot Recall^w}{\beta^2 \cdot Precision^w + Recall^w} \tag{33}$$

where, β signifies the effectiveness of detection with respect to a user who attaches β times as much importance to $Recall^w$ as to $Precision^w$, normally $\beta = 1$. Then, we apply $F_1^w - measure$ to calculate the *RA* of different segmentation results. The higher *RA* is, the better the segmentation achieved is.

4.2. Boundary Accuracy

The *BA* [39] is defined according to the Hausdorff distance. The boundary pixels of ground true image and segmented image are defined as B_{GT} and B_{SEG} as shown in Figure 10.

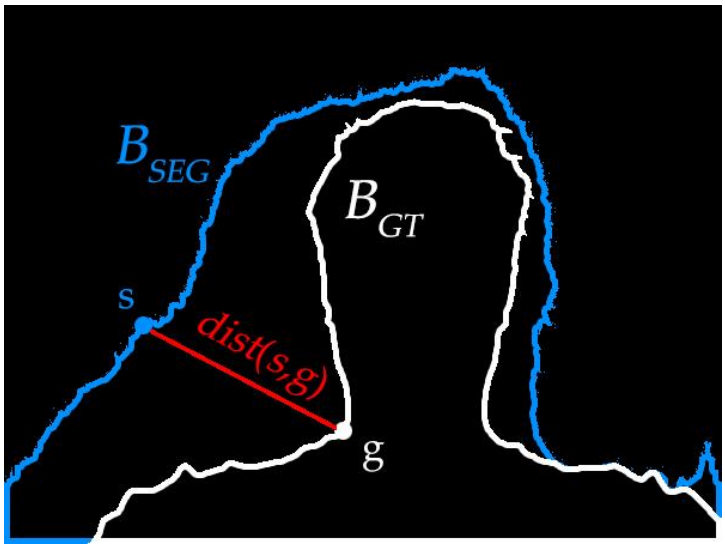


Figure 10. Boundary extraction.

The formula is as follows:

$$BA = \frac{N(B_{SEG}) + N(B_{GT})}{\sum_s \min_g(dist(s,g)) + \sum_g \min_s(dist(g,s))}, \tag{34}$$

where, $g \in B_{GT}$ and $s \in B_{SEG}$, $dist(\cdot)$ denotes the Euclidean distance, $N(\cdot)$ is the pixel number in the set. The value of BA shows the segmentation accuracy of boundaries.

4.3. Results Analysis

We segmented the images from the dataset by graph cut and random walker as shown in Figure 11. The segmentation test of our method has been made on Gulshan's dataset as well as our hand gesture images, and some of the results using our method on hand gesture image segmentation are shown here in Figure 12.

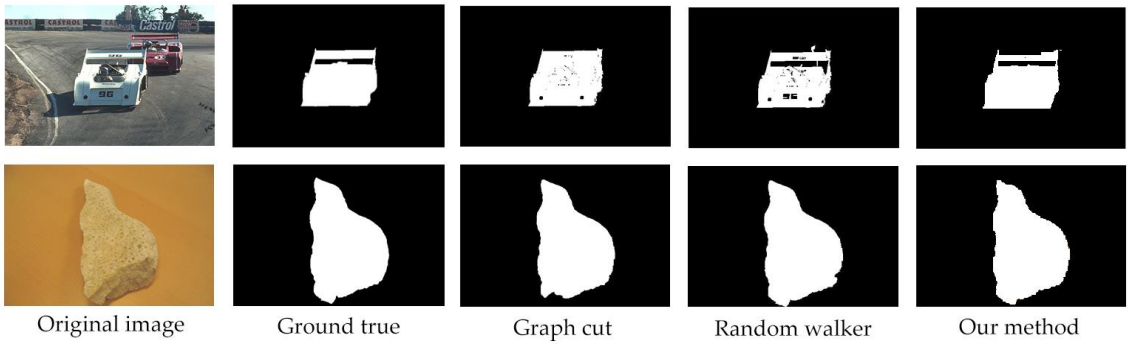


Figure 11. The evaluation on different algorithms.

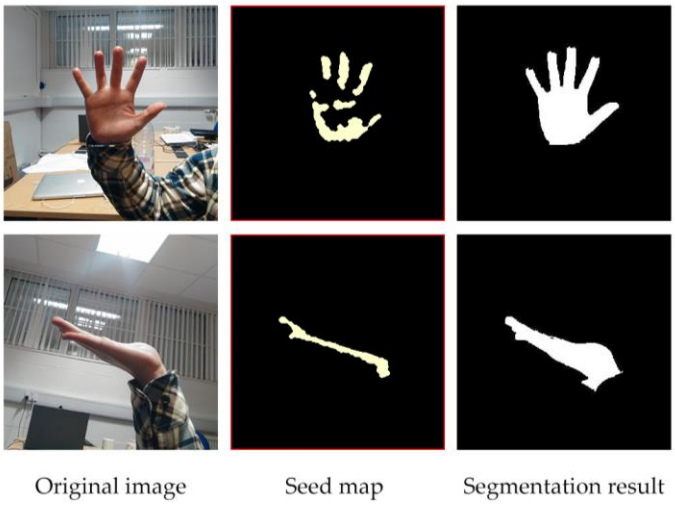


Figure 12. Segmentation results of our method on hand images.

For a more rigorous test, we tested 151 images from Gulshan's dataset and used the human interaction simulator to perform the interactions, which generated the seeds 20 times to further refine the segmentation results. The result of each simulation step has been tested on the experiment platform. The RA and BA scores are the mean values of 151 segmentations, shown in Figures 13 and 14.

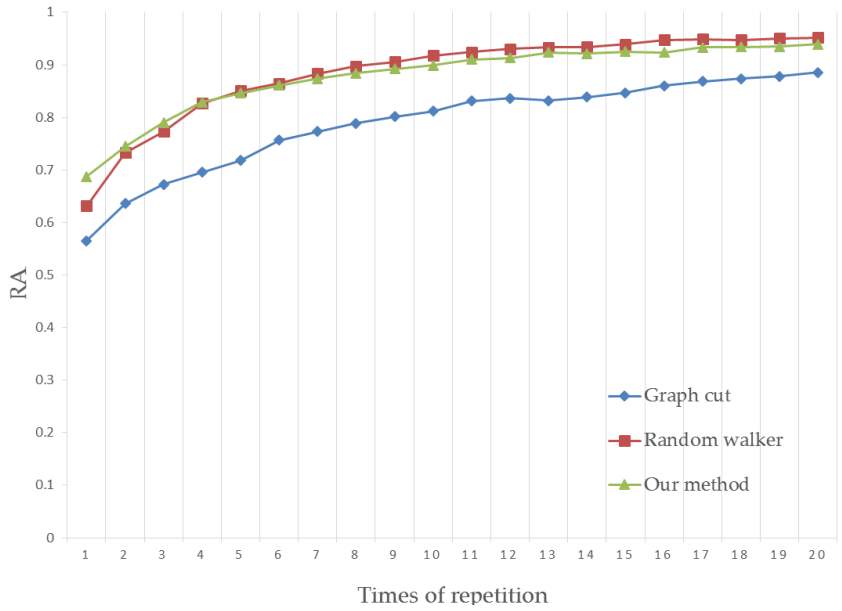


Figure 13. Region accuracy comparison.

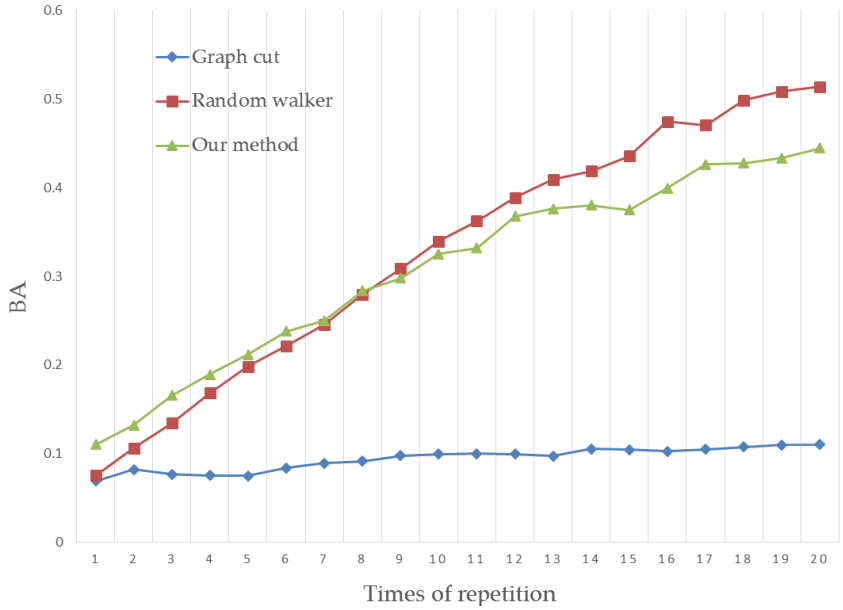


Figure 14. Boundary accuracy comparison.

From the figures above, the segmentation quality shows an increase with simulated human interactions. When the seed number becomes high, a satisfactory segmentation will be achieved. Our method obtains the best segmentation quality with few human interactions. Since the seeds are generated once automatically in human hand image segmentation, our method is suitable for human image segmentation.

5. Hand Gesture Recognition

We defined five hand gestures: hand closed (HC), hand open (HO), wrist extension (WE), wrist flexion (WF), and fine pitch (FP), as shown in Figure 15.

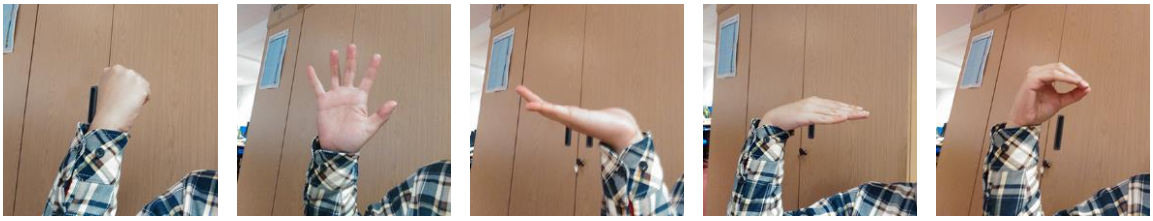


Figure 15. Five hand gestures for recognition.

One hundred images of each hand gesture were captured and segmented by the proposed method. We used the recognition framework in Figure 16. Each gesture takes 50 images for training and 50 for testing. To achieve a better classification, we extract HOG along with Hu invariant moments at the same weights. The K-SVD dictionary training method [40] is used to choose atoms representing [41] all features and reduce the computation costs.

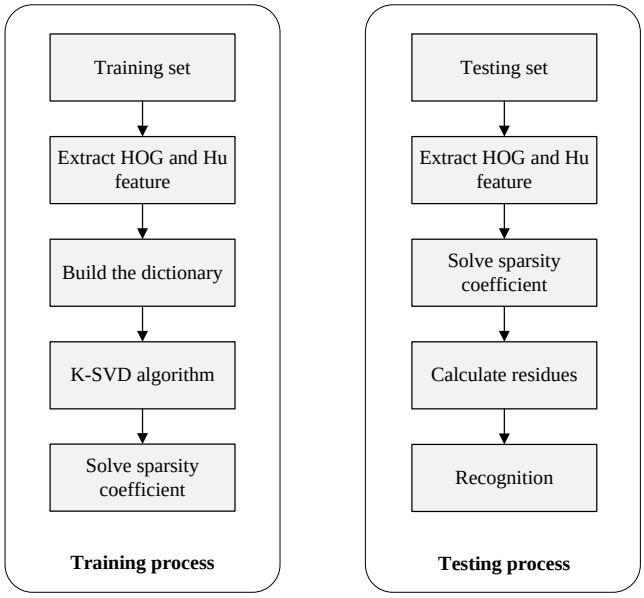


Figure 16. Hand gesture recognition framework.

We tested the recognition rates on both unsegmented hand images and segmented hand images. The recognition rates on unsegmented hand images are shown in Table 2, and the recognition rates on segmented hand images are shown in Table 3.

Table 2. Recognition rates on unsegmented hand images.

Gestures	Recognition Rates
Hand close	86.7%
Hand open	73.3%
Wrist extension	100%
Wrist flexion	100%
Fine pitch	66.7%
Over all rate	**85.3%**

Table 3. Recognition rates on segmented hand images.

Gestures	Recognition Rates
Hand close	93.3%
Hand open	100%
Wrist extension	100%
Wrist flexion	100%
Fine pitch	100%
Over all rate	**98.7%**

By segmenting the images before feature extraction, the recognition rates on those five hand gestures are increased compared with unsegmented images, according to the results in the tables above.

6. Conclusions and Future Work

In conclusion, the interactive hand gesture image segmentation method can perfectly meet the segmentation demands of hand gesture images with no human interactions. The mechanism behind this method is carefully explored and deduced with the assistance of modern mathematical theories. Comparing the segmentation results of hand gestures with other popular image segmentation

methods, our method can obtain a better segmentation accuracy and a higher quality, when there are limited seeds. Automatic seeds selection also helps to reduce human interactions. The segmentation work in turn improves the recognition rate. In future work, we could adapt this method to higher resolution pictures, which requires simplifying the calculation process. In seed selection, the automatic selection method could be improved to overcome various interferes, such as highlights, shadows and image distortion. Other future work will focus on improving the recognition rate by integrating the segmentation algorithm with more advanced recognition methods.

Acknowledgments: This work was supported by grants of National Natural Science Foundation of China (Grant No. 51575407, 51575338, 61273106, 51575412) and the EU Seventh Framework Programme (Grant No. 611391).

Author Contributions: D.C. and G.L. conceived and designed the experiments; D.C. performed the experiments; D.C. and G.L. analyzed the data; D.C. contributed reagents/materials/analysis tools; D.C. wrote the paper; H.L., Z.J. and H.Y. edited the language.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nardi, B.A. *Context and Consciousness: Activity Theory and Human-Computer Interaction*; MIT Press: Cambridge, MA, USA, 1996; p. 400.
2. Chen, D.C.; Li, G.F.; Jiang, G.Z.; Fang, Y.F.; Ju, Z.J.; Liu, H.H. Intelligent Computational Control of Multi-Fingered Dexterous Robotic Hand. *J. Comput. Theor. Nanosci.* **2015**, *12*, 6126–6132. [CrossRef]
3. Ju, Z.J.; Zhu, X.Y.; Liu, H.H. Empirical Copula-Based Templates to Recognize Surface EMG Signals of Hand Motions. *Int. J. Humanoid Robot.* **2011**, *8*, 725–741. [CrossRef]
4. Miao, W.; Li, G.F.; Jiang, G.Z.; Fang, Y.; Ju, Z.J.; Liu, H.H. Optimal grasp planning of multi-fingered robotic hands: A review. *Appl. Comput. Math.* **2015**, *14*, 238–247.
5. Farina, D.; Jiang, N.; Rehbaum, H.; Holobar, A.; Graimann, B.; Dietl, H.; Aszmann, O.C. The extraction of neural information from the surface EMG for the control of upper-limb prostheses: Emerging avenues and challenges. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2014**, *22*, 797–809. [CrossRef] [PubMed]
6. Ju, Z.; Liu, H. Human Hand Motion Analysis with Multisensory Information. *IEEE/ASME Trans. Mechatron.* **2014**, *19*, 456–466. [CrossRef]
7. 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. [CrossRef]
8. Yang, D.F.; Wang, S.C.; Liu, H.P.; Liu, Z.J.; Sun, F.C. Scene modeling and autonomous navigation for robots based on kinect system. *Robot* **2012**, *34*, 581–589. [CrossRef]
9. Wang, C.; Liu, Z.; Chan, S.C. Superpixel-Based Hand Gesture Recognition with Kinect Depth Camera. *Trans. Multimed.* **2015**, *17*, 29–39. [CrossRef]
10. Sinop, A.K.; Grady, L. A Seeded Image Segmentation Framework Unifying Graph Cuts and Random Walker Which Yields a New Algorithm. In Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV), Rio de Janeiro, Brazil, 14–20 October 2007; pp. 1–8.
11. Grady, L. Multilabel random walker image segmentation using prior models. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–26 June 2005; pp. 763–770.
12. Couprie, C.; Grady, L.; Najman, L.; Talbot, H. Power watersheds: A new image segmentation framework extending graph cuts, random walker and optimal spanning forest. In Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV), Kyoto, Japan, 27 September–4 October 2009; pp. 731–738.
13. Varun, G.; Carsten, R.; Antonio, C.; Andrew, B.; Andrew, Z. Geodesic star convexity for interactive image segmentation. In Proceedings of the IEEE CVPR, San Francisco, CA, USA, 13–18 June 2010; pp. 3129–3136.
14. Ju, Z.; Liu, H. A Unified Fuzzy Framework for Human Hand Motion Recognition. *IEEE Trans. Fuzzy Syst.* **2011**, *19*, 901–913.
15. Xu, Y.; Yu, G.; Wang, Y.; Wu, X.; Ma, Y. A Hybrid Vehicle Detection Method Based on Viola-Jones and HOG + SVM from UAV Images. *Sensors* **2016**, *16*, 1325. [CrossRef] [PubMed]
16. Fernando, M.; Wijjayanayake, J. Novel Approach to Use Hu Moments with Image Processing Techniques for Real Time Sign Language Communication. *Int. J. Image Process.* **2015**, *9*, 335–345.

17. Chen, Q.; Georganas, N.D.; Petriu, E.M. Real-time vision-based hand gesture recognition using haar-like features. In Proceedings of the EEE Instrumentation & Measurement Technology Conference IMTC, Warsaw, Poland, 1–3 May 2007; pp. 1–6.
18. Sun, R.; Wang, J.J. A Vehicle Recognition Method Based on Kernel K-SVD and Sparse Representation. *Pattern Recognit. Artif. Intell.* **2014**, *27*, 435–442.
19. Jiang, Y.V.; Won, B.-Y.; Swallow, K.M. First saccadic eye movement reveals persistent attentional guidance by implicit learning. *J. Exp. Psychol. Hum. Percept. Perform.* **2014**, *40*, 1161–1173. [CrossRef] [PubMed]
20. Ju, Z.; Liu, H.; Zhu, X.; Xiong, Y. Dynamic Grasp Recognition Using Time Clustering, Gaussian Mixture Models and Hidden Markov Models. *Adv. Robot.* **2009**, *23*, 1359–1371. [CrossRef]
21. Bian, X.; Zhang, X.; Liu, R.; Ma, L.; Fu, X. Adaptive classification of hyperspectral images using local consistency. *J. Electron. Imaging* **2014**, *23*, 063014.
22. Song, H.; Wang, Y. A spectral-spatial classification of hyperspectral images based on the algebraic multigrid method and hierarchical segmentation algorithm. *Remote Sens.* **2016**, *8*, 296. [CrossRef]
23. Hatwar, S.; Anil, W. GMM based Image Segmentation and Analysis of Image Restoration Tecniques. *Int. J. Comput. Appl.* **2015**, *109*, 45–50. [CrossRef]
24. Couprie, C.; Najman, L.; Talbot, H. Seeded segmentation methods for medical image analysis. In *Medical Image Processing*; Springer: New York, NY, USA, 2011; pp. 27–57.
25. Bańbura, M.; Modugno, M. Maximum likelihood estimation of factor models on datasets with arbitrary pattern of missing data. *J. Appl. Econ.* **2014**, *29*, 133–160. [CrossRef]
26. Simonetto, A.; Leus, G. Distributed Maximum Likelihood Sensor Network Localization. *IEEE Trans. Signal Process.* **2013**, *62*, 1424–1437. [CrossRef]
27. Ju, Z.; Liu, H. Fuzzy Gaussian Mixture Models. *Pattern Recognit.* **2012**, *45*, 1146–1158. [CrossRef]
28. Zhang, Y.; Brady, M.; Smith, S. Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. *IEEE Trans. Med. Imaging* **2001**, *20*, 45–57. [CrossRef] [PubMed]
29. Song, W.; Cho, K.; Um, K.; Won, C.S.; Sim, S. Intuitive terrain reconstruction using height observation-based ground segmentation and 3D object boundary estimation. *Sensors* **2012**, *12*, 17186–17207. [CrossRef] [PubMed]
30. Wei, S.; Kyungeun, C.; Kyhyun, U.; Chee, S.; Sungdae, S. Complete Scene Recovery and Terrain Classification in Textured Terrain Meshes. *Sensors* **2012**, *12*, 11221–11237.
31. Liao, L.; Lin, T.; Li, B.; Zhang, W. MR brain image segmentation based on modified fuzzy C-means clustering using fuzzy GIbbs random field. *J. Biomed. Eng.* **2008**, *25*, 1264–1270.
32. Kakumanu, P.; Makrogiannis, S.; Bourbakis, N. A survey of skin-color modeling and detection methods. *Pattern Recognit.* **2007**, *40*, 1106–1122. [CrossRef]
33. Lee, G.; Lee, S.; Kim, G.; Park, J.; Park, Y. A Modified GrabCut Using a Clustering Technique to Reduce Image Noise. *Symmetry* **2016**, *8*, 64. [CrossRef]
34. Ning, J.; Zhang, L.; Zhang, D.; Wu, C. Interactive image segmentation by maximal similarity based region merging. *Pattern Recognit.* **2010**, *43*, 445–456. [CrossRef]
35. Grabcut Image Dataset. Available online: http://research.microsoft.com/enus/um/cambridge/projects/visionimagevideoediting/segmentation/grabcut.htm (accessed on 18 December 2016).
36. Everingham, M.; Van, G.L.; Williams, C.K.; Winn, I.J.; Zisserman, A. The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results. Available online: http://host.robots.ox.ac.uk/pascal/VOC/voc2009/ (accessed on 26 December 2016).
37. Rhemann, C.; Rother, C.; Wang, J.; Gelautz, M.; Kohli, P.; Rott, P. A perceptually motivated online benchmark for image matting. In Proceedings of the CVPR, Miami, FL, USA, 20–25 June 2009; pp. 1826–1833.
38. Margolin, R.; Zelnik-Manor, L.; Tal, A. How to Evaluate Foreground Maps? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 248–255.
39. Zhao, Y.; Nie, X.; Duan, Y. A benchmark for interactive image segmentation algorithms. In Proceedings of the IEEE Person-Oriented Vision, Kona, HI, USA, 7 January 2011; pp. 33–38.

40. Zhou, Y.; Liu, K.; Carrillo, R.E.; Barner, K.E.; Kiamilev, F. Kernel-based sparse representation for gesture recognition. *Pattern Recognit.* **2013**, *46*, 3208–3222. [CrossRef]
41. Yu, F.; Zhou, F. Classification of machinery vibration signals based on group sparse representation. *J. Vibroeng.* **2016**, *18*, 1540–1545. [CrossRef]