

# Intelligent Spot Detection for 2-DE Gel Image<sup>\*</sup>

Yi-Sheng Liu<sup>1</sup>, Shu-Yuan Chen<sup>1</sup>, Ya-Ting Chao<sup>1</sup>, Ru-Sheng Liu<sup>1</sup>,  
Yuan-Ching Tsai<sup>2</sup>, and Jaw-Shu Hsieh<sup>2</sup>

<sup>1</sup> Department of Computer Science and Engineering, Yuan Ze University,  
Chung Li, Taiwan

s889404@mail.yzu.edu.tw, {cschen, ytchao, csrobinl}@saturn.yzu.edu.tw

<sup>2</sup> Department of Agronomy, National Taiwan University, Taipei, Taiwan  
botsai@gate.sinica.edu.tw, jawar@ntu.edu.tw

**Abstract.** In this study, a novel method for spot detection is proposed with the addition of confidence evaluation for each detected spot. The confidence of a spot will give useful hints for subsequent processing such as landmark selection, spot quantification, gel image registration, etc. The proposed method takes slices of a gel image in the gray level direction and build them into a slice tree, which in turn is used to perform spot detection and confidence evaluation. Moreover, the proposed method is fast. Building slice tree for a gel image of  $1262 \times 720$  take about 3.2 sec. Spot detection take about 66 ms after the slice tree was built. Experimental results show that confidence values are close to subjective judgement.

**Keywords:** 2-DE gel, Protein, Spot detection, Slice tree.

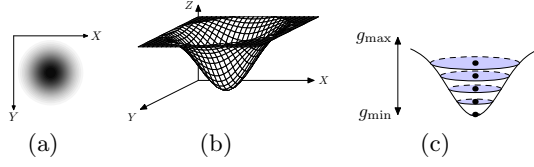
## 1 Introduction

*Proteomics* is the study of proteome, especially how the proteins are functioning in and around cells. Protein separation is one of the most important stages in the proteomics study. Among all separation techniques, two-dimensional electrophoresis (2-DE) [1,2,3,4] is the best method to separate complex protein mixtures according to their charge and size. Spots in the gel are proteins migrated to specific locations. According to the differential expression of protein mixtures from control and experimental samples, the spots in gel may disappear, appear or change in size and intensity. By analysis of spot appearance in gel, differential protein expression between various samples are obtained.

Due to the volume data and technical noise originating from the image acquisition process, manual analysis of gel image is difficult without the help of computer software. Analysis of gel image by image processing software requires an image pipeline which may contain image correction, spot detection, spot

---

<sup>\*</sup> This work was partially supported by the National Science Council of Taiwan, R.O.C., under Grants NSC-95-2745-E-155-008-URD.



**Fig. 1.** Slices of spot. (a) A spot in gel image. (b) A spot in three dimension view. (c) Slices of spot and corresponding central points.

quantification, spot registration, data presentation and interpretation. Quality of spot detection is one of the most important factors that will influence the performance of the image pipeline. A comprehensive review of those computation techniques can be referred to [3].

In this study, a novel method for spot detection is proposed with the addition of confidence calculation for each detected spot. The confidence of a spot will give useful hints for subsequent processing such as landmark selection, spot matching, gel image registration, etc. The proposed method takes slices of a gel image in the gray level direction and build them into a slice tree, which in turn is used to perform spot detection and confidence calculation.

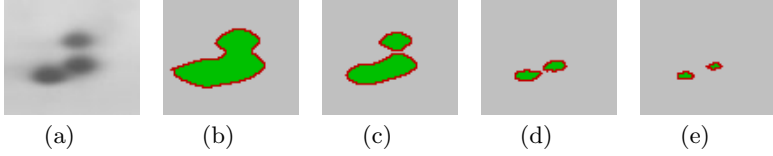
This paper is organized as follows. The idea to detect spots in gel images by slice tree is presented in Section 2. The detailed description about our algorithm is presented in Section 3. Finally, experimental results and conclusions are given in Sections 4 and 5, respectively.

## 2 Approach

The key concept of the proposed approach is introduced in this section. Unlike other spot detection method, our method slices the gel image, builds them into a slice tree, and then detects spots on the basis of slice tree.

Let the intensity be the third-dimension ( $Z$  axis), the intensity of a sliver-stained spot is approximately Gaussian distributed with the lowest intensity at center as shown in Fig. 1(b). A series of slices of the spot can then be obtained in the intensity direction as shown in Fig. 1(c). Each slice has its own features such as size, shape, central point, boundary smoothness and so on. If we project the central points of slices onto the  $X$ - $Y$  plan. The projected points belong to the same spot will fall in a neighborhood. The distribution and the number of projected points depend on the shape and appearance of the spots in a gel image which can be used for spot detection.

In fact, spots may be distorted [1,5], overlapping [6] and suffered with noise. These factors make spot detection more difficult and un-reliable. The relationship between the slices of the spots can be incorporated into the slice tree so as to resolve these problems and then obtain a robust spot detector.



**Fig. 2.** Border tracing to detect regions in binary images. (a) Sample gel image with gray levels from 90 to 218. (b)-(e) are detected results of  $B_{187}$ ,  $B_{162}$ ,  $B_{108}$  and  $B_{96}$  respectively. A border is a contour labeled by red color and all the pixels with green color inside the border compose the corresponding region.

### 3 Methods

#### 3.1 Gel Image Slicing

For a 2D-gel image  $I$ , the binarized image  $B_g$  related to gray level  $g$  is defined by

$$B_g(x, y) = \begin{cases} 1 & \text{if } I(x, y) \leq g, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where  $I(x, y)$  is the intensity of pixel at coordinates  $(x, y)$  and  $g$  is one of the gray levels between the maximum and minimum gray levels of  $I$ , denoted by  $g_{\max}$  and  $g_{\min}$ , respectively.

**Definition 1.** *Regions.* Let  $r$  be a subset of pixels in a binary image. We call  $r$  a region in a binary image if  $r$  is a connected set.

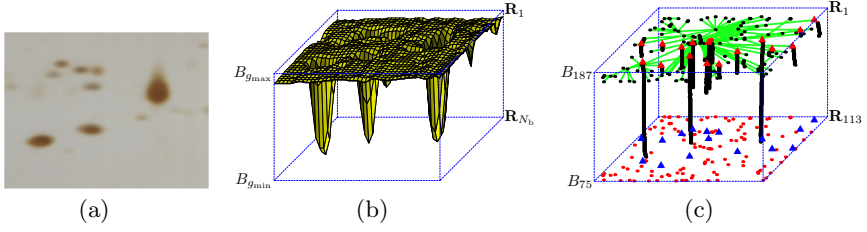
Regions can be detected in the following way. Region borders in a binary image are first detected by border tracing. The set of pixels enclosed by a border is then denoted as the corresponding region. Some results of border tracing and the detected regions are shown in Fig. 2. From this figure, we can also find that a candidate spot with minimum gray level  $g_{s_{\min}}$  will appear as sequence of regions in binary images  $B_{g_s}$  for  $g_{\max} \geq g_s \geq g_{s_{\min}}$ . Intuitively, the sequence of binary images from  $B_{g_{\max}}$  to  $B_{g_{\min}}$  can be regarded as computerized tomography (CT) images of all the spots in the gray level direction, i.e.  $Z$  axis.

**Definition 2.** *Region set.* All regions in a binary image is called a region set of the binary image.

For the gel image  $I$ , there are  $N_b = g_{\max} - g_{\min} + 1$  binary images. We sort binary images  $B_g$  in the descending order of  $g$  and let  $\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_{N_b}$  be the region sets related to the binary images  $B_{g_{\max}}, B_{g_{\max}-1}, \dots, B_{g_{\min}}$ , respectively. i.e.

$$\mathbf{R}_s = \{r_{s,i} | i = 1, 2, \dots, n_s\}, s = 1, 2, \dots, N_b \quad (2)$$

where  $r_{s,i}$  are the regions in binary image  $B_{g_{\max}+1-s}$  and  $n_s$  is the number of regions in the binary image  $B_{g_{\max}+1-s}$ . Note that the value of  $s$  can be regarded as the layer index of the region sets. A sample gel image to illustrate the relationship between  $B_g$  and  $\mathbf{R}_s$  is shown in Fig. 3(b).



**Fig. 3.** Slice tree of gel image. (a) Sample gel image. (b) 3D view of (a), relationship between  $B_g$  and  $\mathbf{R}_s$  are shown beside the cube. (c) Corresponding slice tree of (a). Central points of regions are shown by black circular points, parent and children region are connected by green links, detected spots are shown by triangular marks. Projections of region centers and spots are also shown in  $B_{gmin}$ .

### 3.2 Properties of Regions

Some properties of regions are described in this section.

**Definition 3.** *Binary image projection.* For a binary image  $B_g$ , the projection of  $B_g$ ,  $\Psi(B_g)$ , is defined as a set of coordinates whose corresponding pixel values are 1. i.e.

$$\Psi(B_g) = \{(x, y) | B_g(x, y) = 1\} . \quad (3)$$

Since a region is a subset of a binary image, the operation  $\Psi$  can also be applied to a region. i.e.

$$\Psi(r_{s,i}) = \{(x, y) | r_{s,i}(x, y) = 1\} . \quad (4)$$

**Definition 4.** *Ancestor region and descendant region.* For two regions  $r_{s_1,i}$  and  $r_{s_2,j}$ , with  $s_1 < s_2$ , if  $\Psi(r_{s_1,i}) \supseteq \Psi(r_{s_2,j})$ , then we say  $r_{s_1,i}$  is ancestor region of  $r_{s_2,j}$  and  $r_{s_2,j}$  is descendant region of  $r_{s_1,i}$ , and denoted by

$$r_{s_1,i} \ni r_{s_2,j} . \quad (5)$$

**Definition 5.** *Child region and parent region.* For two regions  $r_{s_1,i}$  and  $r_{s_2,j}$ , if  $s_1 = s_2 - 1$  and  $r_{s_1,i} \ni r_{s_2,j}$ , then we say  $r_{s_2,j}$  is a child region of  $r_{s_1,i}$  and  $r_{s_1,i}$  is the parent region of  $r_{s_2,j}$ .

*Property 1.* All regions in a binary image are mutual exclusive. i.e.

$$\Psi(r_{s,i}) \cap \Psi(r_{s,j}) = \emptyset \text{ if } i \neq j . \quad (6)$$

*Property 2.* Every region in  $\mathbf{R}_2, \mathbf{R}_3, \dots, \mathbf{R}_{N_b}$  has exactly one parent region. i.e. For  $s = 2, 3, \dots, N_b$

$$\forall r_{s,i} \in \mathbf{R}_s, \exists! r_{s-1,k} \in \mathbf{R}_{s-1}, \text{ s.t. } \Psi(r_{s,i}) \subseteq \Psi(r_{s-1,k}) . \quad (7)$$

*Property 3.* For two regions  $r_{s,i}$  and  $r_{s-1,k}$ ,  $r_{s,i}$  is a child region of  $r_{s-1,k}$  if and only if their projection are overlapping. i.e.

$$\Psi(r_{s,i}) \subseteq \Psi(r_{s-1,k}) \iff \Psi(r_{s,i}) \cap \Psi(r_{s-1,k}) \neq \emptyset . \quad (8)$$

Property 3 will simplify the procedure of finding child regions. For a region  $r_{s,i} \in \mathbf{R}_s$ , the child regions of  $r_{s,i}$  can be found in the area  $\Psi(r_{s,i})$  of binary image  $B_{g_{\max}-s}$ . Let the set of all child regions of  $r_{s,i}$  be denoted by  $\mathbf{R}_{s+1,i}$ , then it is obvious that  $\mathbf{R}_{s+1} = \bigcup_{i=1}^{n_s} \mathbf{R}_{s+1,i}$ . By Property 3,  $\mathbf{R}_{s,i}$  can be constructed as

$$\mathbf{R}_{s,i} = \{r_{s+1,j} | \Psi(r_{s+1,j}) \cap \Psi(r_{s,i}) \neq \emptyset\}, j = 1, 2, \dots, n_{s+1} . \quad (9)$$

### 3.3 Slice Tree

Regions in binary images are the basic units for spot detection and confidence calculation in our method. To increase the robustness of spot detection, the relationship between regions in successive binary images related to the same spot are organized in a slice tree.

**Definition 6.** *Slice tree.* A slice tree for gel image  $I$  can be defined as  $T = (\mathbf{V}, \mathbf{E})$ , where  $\mathbf{V}$  is a set of nodes and  $\mathbf{E}$  is a set of links between the nodes.

Each node in the slice tree corresponds to a region. Hence, the node related to the region  $r_{s,i}$  is denoted as  $\mathcal{V}(r_{s,i})$ . According to the layer structure of region sets in (2),  $\mathbf{V}$  can be further divided into  $N_b$  exclusive subsets, that is

$$\mathbf{V} = \bigcup_{s=1}^{N_b} \mathbf{V}_s . \quad (10)$$

Nodes in  $\mathbf{V}_s$  correspond to regions in  $\mathbf{R}_s$ , i.e.,

$$\mathbf{V}_s = \{\mathcal{V}(r_{s,i}) | i = 1, 2, \dots, n_s\} . \quad (11)$$

Note that nodes in  $\mathbf{V}_s$  have depth  $s - 1$  in the slice tree.

Each link in  $\mathbf{E}$  is an ordered pair of nodes  $(\mathcal{V}(r_{s-1,k}), \mathcal{V}(r_{s,i}))$ , where  $r_{s-1,k}$  is a parent region and  $r_{s,i}$  is a child region, i.e.,

$$\mathbf{E} = \{(\mathcal{V}(r_{s-1,k}), \mathcal{V}(r_{s,i})) | r_{s-1,k} \ni r_{s,i}\}, s = 2, \dots, N_b . \quad (12)$$

The slice tree for the sample gel image in Fig. 3(a) is shown in Fig. 3(c).

### 3.4 Slice Tree Construction

A slice tree for gel image  $I$  is constructed in the sequence of  $B_{g_{\max}}, \dots, B_{g_{\min}}$  accompanied by the establishment of relations between every two successive region sets  $\mathbf{R}_s$  and  $\mathbf{R}_{s+1}$  for  $s = 1, 2, \dots, N_b - 1$ . More specifically, the slice tree is built by recursively performing procedure **ProcessChildSlice**( $r_{s,i}$ ) with parameter as the region  $r_{s,i}$  in  $\mathbf{R}_s$ . The pseudo code of the procedure is outlined as follow.

**Procedure ProcessChildSlice**( $r_{s,i}$ )

1. Get child region set  $\mathbf{R}_{s,i}$  of  $r_{s,i}$  using (9).
2. If  $\mathbf{R}_{s,i} = \emptyset$  then return,  
     else for all child regions  $r_{s+1,j} \in \mathbf{R}_{s,i}$ , do
  - 2.1 Create a tree node  $\mathcal{V}(r_{s+1,j})$ .
  - 2.2 Set parent-child link between  $\mathcal{V}(r_{s,i})$  and  $\mathcal{V}(r_{s+1,j})$ .
  - 2.3 If  $s < N_b - 1$  then call **ProcessChildSlice**( $r_{s+1,j}$ ).

To build slice tree for a gel image, we first create a root node  $\mathcal{V}(r_{1,1})$ , then call **ProcessChildSlice**( $r_{1,1}$ ). From (1),  $r_{1,1} = B_{g_{\max}}$  covers the whole gel image and is the only region in  $\mathbf{R}_1$ .

**3.5 Slice Tree Terminology**

Let  $N(\mathbf{R}_{s,i})$  denotes the number of child regions for  $r_{s,i}$ , then node  $\mathcal{V}(r_{s,i})$  has  $N(\mathbf{R}_{s,i})$  children in the slice tree. Nodes in slice tree can be divided into three categories according to the number of children:

1. Leaf nodes:  $N(\mathbf{R}_{s,i}) = 0$ .
2. Solitary nodes:  $N(\mathbf{R}_{s,i}) = 1$ .
3. Manifold nodes:  $N(\mathbf{R}_{s,i}) > 1$ .

**Definition 7.** *Sticks.* If we remove all links between manifold nodes and their child nodes, the slice tree is divided into subgroups namely sticks.

Note that all nodes in a stick have no more than one link. It is obvious that each spot in the gel image has a corresponding stick in the slice tree.

**Definition 8.** *Stick root, leaf stick, internal stick, sibling sticks, parent stick and child sticks.* The node with minimum depth in a stick is called stick root. A stick is called a leaf stick if it contains leaf node, otherwise the stick is called an internal stick. Sticks whose stick roots have the same parent node in the original slice tree are called sibling sticks and also called child sticks of the stick where the parent node resides, which in turn is called parent stick.

**Definition 9.** *Stick length.* For a node  $\mathcal{V}(r_{s,i})$ , the stick length of  $\mathcal{V}(r_{s,i})$  is defined as

$$\mathcal{L}(r_{s,i}) = 1 - s + \arg \min_{d=s}^{N_b} (N(\mathbf{R}_{d,i_d}) \neq 1) \quad . \quad (13)$$

Note that for the case of  $d = s$ , the value of  $i_d$  is just  $i$ . For the other cases of  $d$ ,  $i_d$  is an index such that  $r_{d-1,i_{d-1}} \supseteq r_{d,i_d}$ . It is obvious that the stick length of a stick root is equal to the number of nodes in the stick and stick length of each leaf node is 1.

**Definition 10.** *Extended stick length.* For a node  $\mathcal{V}(r_{s,i})$ , the extended stick length of  $\mathcal{V}(r_{s,i})$  is defined as

$$\mathcal{L}_e(r_{s,i}) = 1 - s + \arg \max_{d=s}^{N_b} (N(\mathbf{R}_{d,i_d}) = 0) \quad . \quad (14)$$

The values of  $i_d$  are defined as those in (13). It is obvious that  $\mathcal{L}(r) = \mathcal{L}_e(r)$  if  $\mathcal{V}(r)$  is a node in a leaf stick. Stick length of a stick is defined as the stick length of its stick root.

### 3.6 Spot Detection

Human recognize spots of gel image by size, shape and intensity variation of the spots. To utilize slice tree for spot detection, region size and stick length are used. The region size is expressed by the number of pixels in the region. If a region belong to a spot, it should have reasonable region size. Thus region size should be restricted to reduce image noises in spot detection. Stick length of each node in the slice tree corresponds to intensity gradient of the spot in gel image. A confident spot should have larger stick length but a faint spot has smaller stick length in the slice tree.

More specifically, spot detection using slice tree is done by performing two recursive procedures **FindSpotInTree**( $r_{s_r,i}$ ) and **ProcessStick**( $r_{s,j}$ ) where  $\mathcal{V}(r_{s_r,i})$  is a stick root while  $r_{s,j}$  could be  $r_{s_r,i}$  or a descendant region of  $r_{s_r,i}$ . Three thresholds  $d$ ,  $w_t$  and  $h_t$  are involved in the spot detection where  $d$  is the minimum stick length of a node to be recognized as a spot and  $w_t$  and  $h_t$  are the minimum width and height of a region which could be processed for spot detection. The sensitivity of spot detection can be controlled by adjusting the three threshold values. The pseudo code for spot detection is outlined as follow.

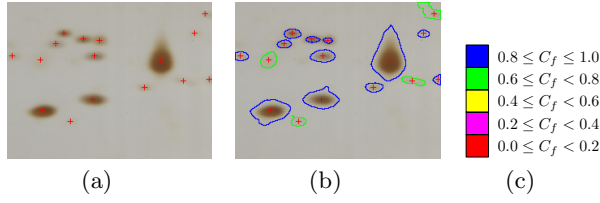
#### Procedure **FindSpotInTree**( $r_{s_r,i}$ )

1. Call **ProcessStick**( $r_{s_r,i}$ ).
2. If no spots are found in all child sticks of  $\mathcal{V}(r_{s_r,i})$  and  $\mathcal{L}_e(r_{s_r,i})$  is greater than or equal to  $d$ , then a spot is found at  $\Psi(r_{s_r,i})$ .

#### Procedure **ProcessStick**( $r_{s,j}$ )

1. If width of  $r_{s,j} \geq w_t$  and height of  $r_{s,j} \geq h_t$  then initiate  $\mathcal{L}(r_{s,j}) = 1$ , else initiate  $\mathcal{L}(r_{s,j}) = 0$ .
2. If  $N(\mathbf{R}_{s,j}) = 1$  then do
  - 2.1 For  $r_{s+1,k} \in \mathbf{R}_{s,j}$ , Call **ProcessStick**( $r_{s+1,k}$ ).
  - 2.2 Set  $\mathcal{L}(r_{s,j}) = \mathcal{L}(r_{s,j}) + \mathcal{L}(r_{s+1,k})$ .
  - 2.3 Goto step 4.
3. If  $N(\mathbf{R}_{s,j}) > 1$  then
  - for all  $r_{s+1,k} \in \mathbf{R}_{s,j}$ , call **FindSpotInTree**( $r_{s+1,k}$ ).
4.  $\mathcal{L}_e(r_{s,j}) = \mathcal{L}(r_{s,j}) + \max_{r \in \mathbf{R}_{s,j}}(\mathcal{L}_e(r))$ .

Parameters  $r_{s_r,i}$  passed to **FindSpotInTree**( $r_{s_r,i}$ ) are regions corresponding to stick roots. **FindSpotInTree**( $r_{s_r,i}$ ) calls **ProcessStick**() to calculate stick length of  $\mathcal{V}(r_{s_r,i})$  and check spot criteria for the node. If  $\mathcal{V}(r_{s_r,i})$  belongs to a leaf stick and its stick length is greater than or equal to  $d$ , then a spot is found at  $\Psi(r_{s_r,i})$ . If no stick roots of sibling sticks satisfy the criteria, shorter sticks are pruned and the longest stick is merged with the parent stick, which in turn is



**Fig. 4.** Results of spot detection using slice tree. (a) The detected spots were marked with red crosses. (b) Confidence of detected spots are shown in various colors. (c) The mapping between confidence values and colors.

used for criteria testing. The pruning and merging procedure is repeated until a merged stick satisfies the criteria or the root node is reached.

**ProcessStick**( $r_{s,j}$ ) checks the region size of  $r_{s,j}$  and calculates stick length for node  $\mathcal{V}(r_{s,j})$  by recursively calling itself with child region as parameter until a non-solitary node encountered. Those regions smaller than a specified size are eliminated during the calculation of stick length. When a manifold node is encountered, **FindSpotInTree**() is called to check spot criteria for child sticks originated from the manifold node. The results of spot detection for the gel image in Fig. 3(a) are shown in Fig. 4.

### 3.7 Confidence Evaluation for Spots

Since spots in the gel image have specific characteristics in the slice tree, their confidence can be computed by the features of the corresponding regions. More specifically, the confidence values of spots are computed on the basis of slice tree by the following equation.

$$C_f = \frac{\sqrt{(\alpha l)^2 + (\beta s)^2 + (\gamma c)^2}}{\alpha + \beta + \gamma} \quad (15)$$

where  $l$ ,  $s$  and  $c$  are metrics for stick length, smoothness and compactness related to the spots, and  $\alpha$ ,  $\beta$  and  $\gamma$  are the respective weighting factors. If we identify spots by the regions where the spots have been detected, then the metrics are defined as follow.

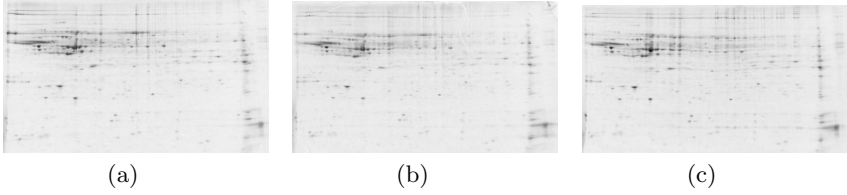
$$l = \min(1.0, \frac{l_e}{\sqrt{n_p}}) \quad (16)$$

$$s = \max(0.0, 1.0 - \delta \times \frac{n_r}{n_b}) \quad (17)$$

$$c = \min(1.0, \frac{2\sqrt{\pi n_p}}{n_b}) \quad (18)$$

where  $l_e$  is the extended stick length related to the spot,  $n_p$  is the number of region pixels,  $n_b$  is the number of border pixels of the region,  $n_r$  is the number of one-pixel-width knobs extended from the region and  $\delta$  is a constant factor. The metrics are normalized to the range from 0 to 1. The larger are the metrics, the more confident spots are obtained.

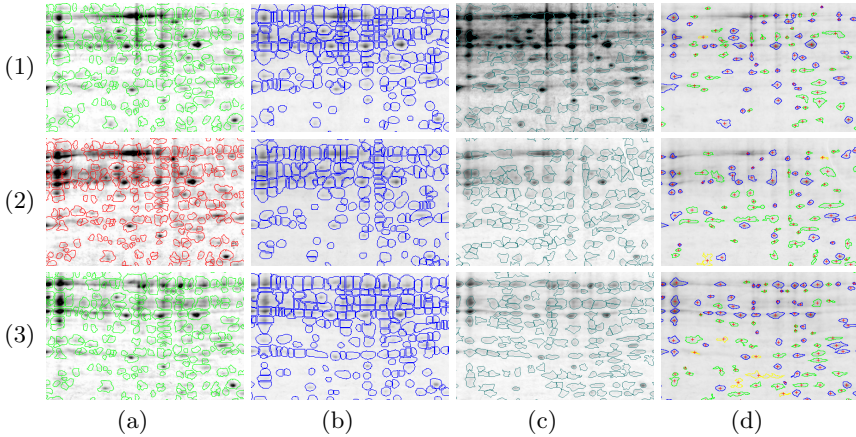




**Fig. 5.** Gel images [7] used for performance evaluation. (a) 031403-ctrl2.tiff ( $1262 \times 724$ ) (b) 031403-ctrl3.tiff ( $1262 \times 720$ ) (c) 031403-ctrl4.tiff ( $1262 \times 700$ )

## 4 Experimental Results

The proposed method has been implemented on a notebook equipped with Intel Pentium III CPU 1.2GHz, 256MB RAM. The gel images [7] used for performance evaluation are shown in Fig. 5. The input gel images are first pre-processed by a  $7 \times 7$  Gaussian filter. The average time took to build slice tree and detect spots for these images is 3.2 sec and 66 ms, respectively.



**Fig. 6.** Comparison of spot detection. Columns for (a) Delta2D, (b) Progenesis, (c) Proteomweaver and (d) Our method. (The mapping between confidence values and colors are as specified in Fig. 4(c)) Rows for (1) 031403-ctrl2.tiff, (2) 031403-ctrl3.tiff and (3) 031403-ctrl4.tiff.

To show the performance of spot detection using slice tree, the results of the proposed spot detection are compared to those of three commercial software [7]: Delta2D 3.2, Progenesis Discovery v.2005 and Proteomweaver 3.0.1.1. Most of the existing spot detection methods including the three methods adopted Watershed [8] algorithm for spot segmentation. After a gel image is segmented, spot models are employed to eliminate segments not being fitted by the model. Watershed is the most popular technique for spot segmentation, over segmentation

is its well-known problem. Thus, the effectiveness of spot model are crucial to the results of spot detection based on watershed. Subblocks of  $429 \times 279$  from the results of spot detection of the software packages are shown in the first three columns of Fig. 6.

Unlike other approaches, spot detection using slice tree does not rely on spot models. Instead, the stick length of each leaf stick corresponding to intensity difference between spots and background is used as the criterion of being confident spot. Results of our method are shown in the fourth column of Fig. 6. In our results, spot centers are marked with red crosses and the boundaries of spots are shown in different color according to their confidence values. It can be seen that the boundaries of spots in our method are more compact.

## 5 Conclusion

Slice tree is effective for representing a gel image in a systematic organization. Nodes in slice tree contain refined features about the spots and links between nodes contain corresponding characteristic expression of the gel image. Thus, gel image analysis can be done by analyzing the slice tree based on the systematic organization. In this paper, we have shown how to detect spots using slice tree. In addition slice tree with confidence evaluation can provide plentiful information for other applications such as spot quantification, gel image registration, etc. These will be the future research issues.

## References

1. Aittokallio, T., Salmi, J., Nyman, T.A., Nevalainen, O.S.: Geometrical distortions in two-dimensional gels: applicable correction methods. *Journal of Chromatography B* **815** (2005) 25–37
2. Salmi, J., Aittokallio, T., Nyman, T.A., Nevalainen, O.S.: Correcting distortions in 2D-gels — a survey. Technical Report 653, Turku Centre for Computer Science (2004)
3. Dowsey, A.W., Dunn, M.J., Yang, G.Z.: The role of bioinformatics in two-dimensional gel electrophoresis. *Proteomics* **3** (2003) 1567–1596
4. Quadroni, M., James, P.: Proteomics and automation. *Electrophoresis* **20** (1999) 664–677
5. Kriegel, K., Seefeldt, I., Hoffmann, F., Schultz, C., Wenk, C., Regitz-Zagrosek, V., Oswald, H., Fleck, E.: An alternative approach to deal with geometric uncertainties in computer analysis of two-dimensional electrophoresis gels. *Electrophoresis* **21** (2000) 2637–2640
6. Pietrogrande, M.C., Marchetti, N., Dondi, F., Righetti, P.G.: Spot overlapping in two-dimensional polyacrylamide gel electrophoresis separations: A statistical study of complex protein maps. *Electrophoresis* **23** (2002) 283–291
7. Dunsmore, J.: Comparison of 2d gel spot detection algorithms. <http://www.deltastat.org/2d-gel-algorithms-comparison.html> (2005)
8. Vincent, L., Soille, P.: Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Analysis and Machine Intelligence* **13** (1991) 583–598