

3D SMAP Algorithm

April 11, 2012

Based on the original SMAP paper [1]. This report extends the structure of MSRF into 3D. The prior distribution is modified to satisfy the MRF property. In addition, an iterative strategy is proposed to refine the segmentation.

1 Goal

Segment 3D image volume into regions of distinct statistical behavior.

2 Basic Model

Y is observed 3D image volume containing distinct textures.

X is unobserved field containing the class of each pixels.

S is 3D lattice of points. s is a member of S .

The lower case letters x_s , y_s denote the corresponding deterministic realization.

3 Approach Outline

1. Assuming the behavior of each observed pixel in Y is dependent on a corresponding unobserved label pixels in X .

2. The dependence is specified through $p_{y|x}(y|x)$, which is modeled by Gaussian mixture distribution (GMM) in our experiment. The parameters of GMM is computed using cluster algorithm described by [2].
3. Prior knowledge about the regions will be specified by the prior distribution $p(x)$, which is modeled by multiscale random field (MSRF).
4. Solve following optimization problem which minimize the average cost of an erroneous segmentation.

$$\hat{x} = \arg \min_x E[C(X, x)|Y = y] \quad (1)$$

5. Once the segmentation is obtained, we can re-estimate the GMM parameters for each class using only a subset of the original data, whose members have the same segmentation labels. Therefore, we can iteratively refine the segmentation results until the termination condition is satisfied.

4 Multiscale Random Field

Multiscale Random Field (MSRF) is composed of a series of 3D random fields at varying scales or resolutions. A demonstration picture is shown by Fig 1. At each scale, n , the segmentation or labeling is denoted by the random field $X^{(n)}$, and the set of lattice points is denoted by $S^{(n)}$. In particular, $X^{(0)}$ is assumed to be the finest scale random field with each point corresponding to a single image pixel. Each label at the next coarser scale $X^{(1)}$ then corresponds to **a group of eight points** in the original image. Therefore, the number of points in $S^{(1)}$ is **1/8** the number of points in $S^{(0)}$. Assume the sequence of random fields from coarse to fine scale form a Markov chain. Then

$$\begin{aligned} & P(X^{(n)} = x^{(n)} | X^{(l)} = x^{(l)}, l > n) \\ &= P(X^{(n)} = x^{(n)} | X^{(n+1)} = x^{(n+1)}) \\ &= P_{x^{(n)}|x^{(n+1)}}(x^{(n)}|x^{(n+1)}) \end{aligned} \quad (2)$$

Define the cost function as follows:

$$C_{SMAP}(X, x) = \frac{1}{2} + \sum_{n=0}^L 2^{n-1} C_n(X, x) \quad (3)$$

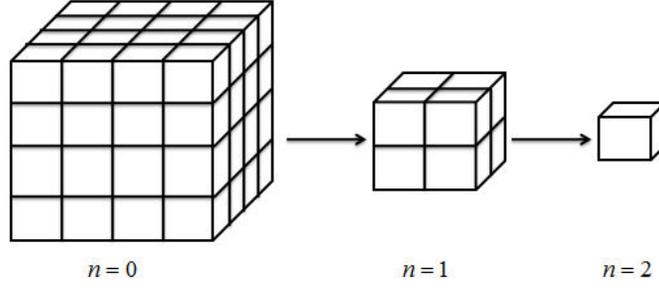


Figure 1: 3D random field with three different scales. Each cube represents a pixel in the 3D image volume

where L is the coarsest scale in X and $C_n(X, x)$ is the penalty function associates with labeling error at the n^{th} scale. Suppose K^{th} scale is the unique scale such that $X^{(K)} \neq x^{(K)}$, and $X^{(i)} = x^{(i)}$ for all $i > K$. Define the functions C_n as

$$C_n(X, x) = \begin{cases} 1, & \text{if } n \leq K \\ 0, & \text{if } n > K \end{cases} \quad (4)$$

Using (3) and (4) We can compute the total cost $C_{SMAP}(X, x) = 2^K$. Intuitively, $C_{SMAP}(X, x)$ can be interpreted as the width of the largest grouping of misclassified pixels.

We can determine the estimator which minimizes the proposed cost function by evaluating (1).

$$\begin{aligned} \hat{x} &= \arg \min_x E[C_{SMAP}(X, x)|Y = y] \\ &= \arg \min_x \sum_{n=0}^L 2^n \{1 - P(X^{(i)} = x^{(i)}, i \geq n|Y = y)\} \\ &= \arg \max_x \sum_{n=0}^L 2^n \{P(X^{(i)} = x^{(i)}, i \geq n|Y = y)\} \end{aligned} \quad (5)$$

Using recursive approach, [1] has proved the following solution for (6)

$$\hat{x}^{(n)} = \arg \max_x \{\log p_{x^{(n)}|x^{(n+1)}, y}(x^{(n)}|\hat{x}^{(n+1)}, y) + \epsilon(x^{(n)})\} \quad (6)$$

where ϵ is a second order term which vanishes to 0 as number of pixels N increase. i.e. $\lim_{N \rightarrow \infty} \epsilon(x^{(n)}) = 0$.

Ignoring ϵ when N is large results in the following recursive equations:

$$\hat{x}^{(L)} = \arg \max_{x^{(L)}} \log p_{x^{(L)}|y}(x^{(L)}|y) \quad (7)$$

$$\hat{x}^{(n)} = \arg \max_{x^{(n)}} \{\log p_{x^{(n)}|x^{(n+1)},y}(x^{(n)}|\hat{x}^{(n+1)}, y)\} \quad (8)$$

Applying the Bayes rule, the Markov properties of X , and assuming that $X^{(L)}$ is uniformly distributed, the SMAP recursion may be rewritten in a form which is more easily computed.

$$\hat{x}^{(L)} = \arg \max_{x^{(L)}} \log p_{y|x^{(L)}}(y|x^{(L)}) \quad (9)$$

$$\hat{x}^{(n)} = \arg \max_{x^{(n)}} \{\log p_{y|x^{(n)}}(y|x^{(n)}) + \log p_{x^{(n)}|x^{(n+1)}}(x^{(n)}|\hat{x}^{(n+1)})\} \quad (10)$$

Now we introduce 3D SMAP algorithm to estimate the solution to (9) and (10).

5 3D SMAP algorithm

This algorithm specifies the conditional density $p_{y|x^{(n)}}(y|x^{(n)})$ together with the coarse to fine scale transition densities $p_{x^{(n)}|x^{(n+1)}}(x^{(n)}|\hat{x}^{(n+1)})$. A hybrid model, which incorporate **3D quadtree** and **3D graph** is used to describe the structure of MSRF. The algorithm estimate the parameters of MSRF during the segmentation process.

The computation is based on following assumption and properties of Markov Random Field (MRF).

First, the observed pixels are conditionally independent given their class labels. Then the conditional density function for the image has the form.

$$p_{y|x^{(0)}}(y|x^{(0)}) = \prod_{s \in S^{(0)}} p_{y_s|x_s^{(0)}}(y|x_s^{(0)}) \quad (11)$$

where $p_{y_s|x_s^{(0)}}(\cdot|k)$ is the conditional density function for an individual pixel given the class label k . We use a multivariate Gaussian mixture density in experiments, but other distribution can be applied as well.

Second, the labelings in $X^{(n)}$ will be conditionally independent, given the labelings in $X^{(n+1)}$.

Furthermore, each labeling $X_s^{(n)}$ will only be dependent on a local neighborhood of labelings ∂s at the next coarser scale. That is

$$p_{x^{(n)}|x^{(n+1)}}(x^{(n)}|\hat{x}^{(n+1)}) = \prod_{s \in S^{(n)}} p_{x_s^{(n)}|x_{\partial s}^{(n+1)}}(x_s^{(n)}|\hat{x}_{\partial s}^{(n+1)}) \quad (12)$$

where $p_{x_s^{(n)}|x_{\partial s}^{(n+1)}}$ is the probability density for $x_s^{(n)}$ given its neighbors at the coarser scale $x_{\partial s}^{(n+1)}$.

The choice of neighborhood ∂s is important, which affects the structure of multiscale pyramid as well as the specification of probability density function described earlier in this section.

1) Compute $p_{y|x^{(n)}}(y|x^{(n)})$ with 3D quadtree

The structure of the quadtree is shown in Fig 2. Each point in the pyramid is only dependent on a single point at the coarser scale. According to Hammersley-Clifford theorem, the probability density function can be characterized by Gibbs distribution. We use the following function to model the probability that $X_s^{(n)}$ has class m , given that its parent is of class k .

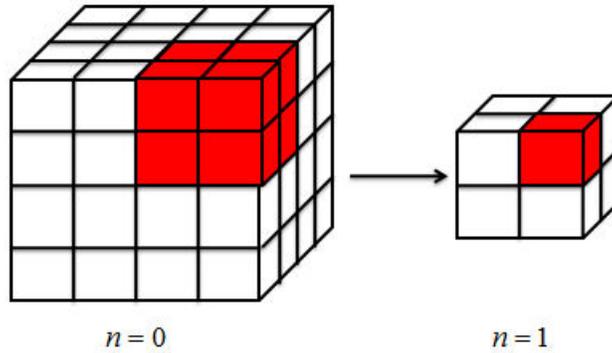


Figure 2: 3D random field with two successive scales. Each cube represents a pixel in the 3D image volume. The red pixels in scale $n = 0$ have the same parent in coarser level $n = 1$, which is also in red.

$$p_{x_s^{(n)}|x_{\partial s}^{(n+1)}}(m|k) = \frac{1}{Z} \exp(-\theta_0^{(n)} \delta_{m \neq k}) \quad (13)$$

where $\delta_{m \neq k} = 1$ if $m \neq k$, otherwise $\delta_{m \neq k} = 0$. $Z = \sum_{m=1}^M \exp\{-\theta_0^{(n)} \delta_{m \neq k}\}$ is called partition function, where M is the number of possible classes. $\theta_0^{(n)} > 0$ is the regularization parameter

of scale n . Large $\theta_0^{(n)}$ results higher probability that the label remain the same from scale $n + 1$ to n .

An important property of the quadtree structure is that the conditional distribution of Y given $X^{(n)}$, has a product form that may be computed using a simple fine-to-coarse recursion. Let $y_s^{(n)}$ be the set of leaves of the 3D quadtree that are on the branch starting at $s \in S^{(n)}$. [1] proves that the conditional density for Y given $X^{(n)}$ has the product form

$$p_{y|x^{(n)}}(y|x^{(n)}) = \prod_{s \in S^{(n)}} p_{y_s^{(n)}|x_s^{(n)}}(y_s^{(n)}|x^{(n)}) \quad (14)$$

Furthermore, the density functions $p_{y_s^{(n)}|x_s^{(n)}}$ may be computed using the following recursion,

$$p_{y_s^{(n+1)}|x_s^{(n+1)}}(y_s^{(n+1)}|k) = \prod_{r \in d^{-1}(s)} \sum_{m=1}^M p_{y_r^{(n)}|x_r^{(n)}}(y_r^{(n)}|m) p_{x_r^{(n)}|x_{\partial r}^{(n+1)}}(m|k) \quad (15)$$

where $d^{-1}(s)$ is a set of points whose first successive parent is s .

Consider the dynamic range, we take the log likelihood function defined as follows.

$$l_s^{(n)}(k) \triangleq \log p_{y_s^{(n)}|x_s^{(n)}}(y_s^{(n)}|k) \quad (16)$$

Substitute the transition distribution of (13) into (15) and converting to log likelihood functions yields the new recursion

$$\begin{aligned} l_s^{(0)}(k) &= \log p_{y_s^{(0)}|x_s^{(0)}}(y_s^{(0)}|k) \\ l_s^{(n+1)}(k) &= \sum_{r \in d^{-1}(s)} \log \left\{ \sum_{m=1}^M \frac{1}{Z} \exp(l_r^{(n)}(m) - \theta_0^{(n)} \delta_{m \neq k}) \right\} \end{aligned} \quad (17)$$

Using (17), we can rewrite the first term of right-hand-side of (10) as

$$\log p_{y|x^{(n)}}(y|x^{(n)}) = \sum_{s \in S^{(n)}} l_s^{(n)}(x_s^{(n)}) \quad (18)$$

2) Compute $p_{x^{(n)}|x^{(n+1)}}(x^{(n)}|\hat{x}^{(n+1)})$ with 3D graph

A disadvantage of the 3D quadtree model is that pixels that are spatially adjacent may not have common neighbors at the next coarser scale. In order to compensate the weakness of discontinuity of boundary, we introduce **3D graph** to model the transition probability

$$p_{x^{(n)}|x^{(n+1)}}(x^{(n)}|\hat{x}^{(n+1)}).$$

The main idea is to increase the number of coarse scale neighbors for each pixels. According to the description in 1), point s has only one coarse scale neighbor, which is the parent node in the 3D quad tree. Now we expand the number of coarse scale neighbors from one to four as shown in Fig. 3. In order to express the positions of the neighbors, we explicitly denote a pixel at scale D as $s = (t, i, j)$, where t, i, j are the frame, row and column indices. The **four neighbors** at scale $D + 1$ may then be computed using the function $odd(i) = 1$ if i is odd and -1 if i is even and the greatest smaller integer function $\lfloor \cdot \rfloor$.

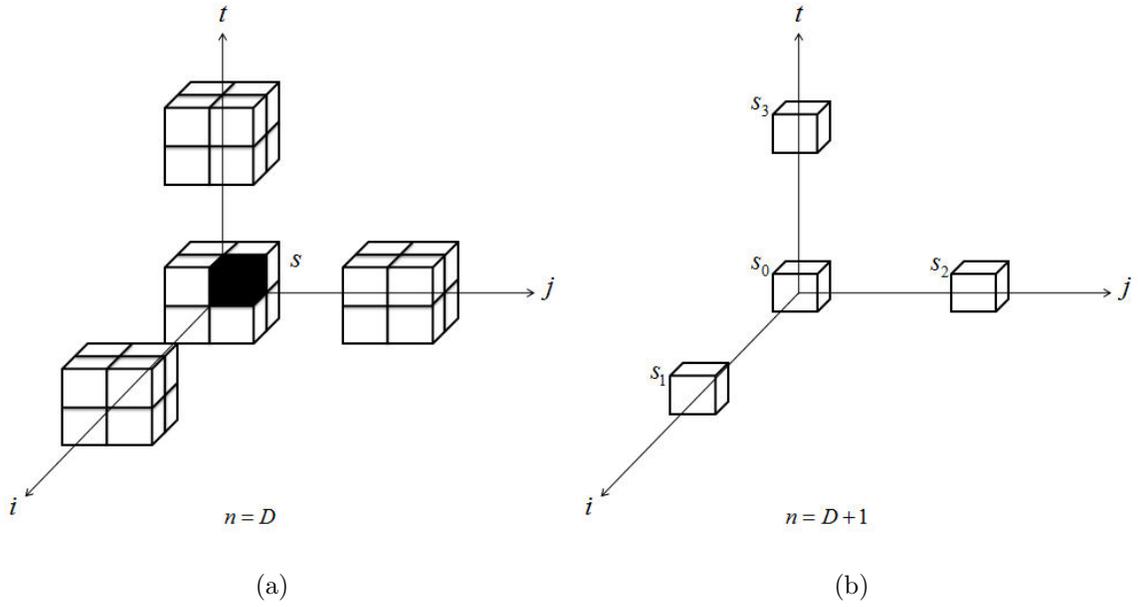


Figure 3: 3D random field with two successive scales. Each cube represents a pixel in the 3D image volume. a) shows 32 pixels in level D , b) shows 4 pixels in coarser level $D + 1$ corresponding to a). In particular, pixel s in level D has four coarser level neighbors: s_0, s_1, s_2, s_3 .

$$\begin{aligned} s_0 &= (\lfloor t/2 \rfloor, \lfloor i/2 \rfloor, \lfloor j/2 \rfloor) \\ s_1 &= (\lfloor t/2 \rfloor, \lfloor i/2 \rfloor, \lfloor j/2 \rfloor) + (odd(t), 0, 0) \\ s_2 &= (\lfloor t/2 \rfloor, \lfloor i/2 \rfloor, \lfloor j/2 \rfloor) + (0, odd(i), 0) \\ s_3 &= (\lfloor t/2 \rfloor, \lfloor i/2 \rfloor, \lfloor j/2 \rfloor) + (0, 0, odd(j)) \end{aligned}$$

The transition function which we choose for this pyramid graph is a natural extension of the transition function used for the quadtree based model. In order to distinguish (13) from the following formula, we use \tilde{p} instead.

$$\begin{aligned}
& \tilde{p}_{x_s^{(n)}|x_{\partial s}^{(n+1)}}(m|m_r), \quad r \in d(s) \\
& = P(X_s^{(n)} = m | X_r^{(n+1)} = m_r) \\
& = \frac{1}{Z} \exp\left(-\sum_{r \in d(s)} \beta_r^{(n)} \delta_{m \neq m_r}\right)
\end{aligned} \tag{19}$$

Similar to (13), $Z = \sum_{m=1}^M \exp\{-\sum_{r \in d(s)} \beta_r^{(n)} \delta_{m \neq m_r}\}$, $d(s) = \{s_0, s_1, s_2, s_3\}$ is a set of points contains all the neighbors of s in the next coarser level. $\beta_r^{(n)}$ can be viewed as a weight parameter associated with the neighbor point r in scale $n+1$. The larger the $\beta_r^{(n)}$, the larger the probability that the label of s will be the same as r .

Besides, we can make the transition smoother by adding neighbor points in the same scale. Define $d^0(s) = \{s'_0, \dots, s'_5\}$ be a set containing six neighbor points in the same scale D as s . See Fig. 4 for a visual demonstration. The coordinates of s' are explicitly given by

$$\begin{aligned}
s'_0 &= (t, i, j-1), \quad s'_1 = (t, i, j+1) \\
s'_2 &= (t, i-1, j), \quad s'_3 = (t, i+1, j) \\
s'_4 &= (t-1, i, j), \quad s'_5 = (t+1, i, j)
\end{aligned}$$

Then (19) can be rewritten as

$$\begin{aligned}
& \tilde{p}_{x_s^{(n)}|x_{\partial s}^{(n+1)}, x_{\partial s}^{(n)}}(m|m_r, m_q), \quad r \in d(s), \quad q \in d^0(s) \\
& = P(X_s^{(n)} = m | X_r^{(n+1)} = m_r, X_q^{(n)} = m_q) \\
& = \frac{1}{Z} \exp\left(-\sum_{r \in d(s)} \beta_r^{(n)} \delta_{m \neq m_r} - \sum_{q \in d^0(s)} \beta_q^{(n)} \delta_{m \neq m_q}\right)
\end{aligned} \tag{20}$$

where $Z = \sum_{m=1}^M \exp\{-\sum_{r \in d(s)} \beta_r^{(n)} \delta_{m \neq m_r} - \sum_{q \in d^0(s)} \beta_q^{(n)} \delta_{m \neq m_q}\}$.

For simplicity, we can use identical weight for points in the same scale, i.e. $\forall r, \beta_r^{(n)} \equiv \theta_1^{(n)}$, $\forall q, \beta_q^{(n)} \equiv \theta_2^{(n)}$.

Using this hybrid pyramid structure of MSRF, the conditional likelihood of (8) and (10) can

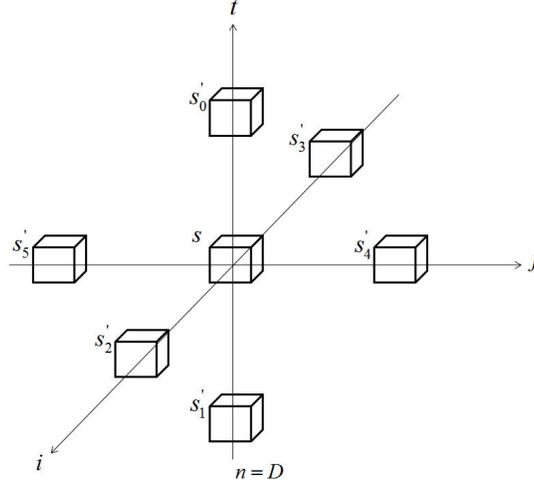


Figure 4: 3D random field of one scale. Each cube represents a pixel in the 3D image volume. s is a pixel in level D with six neighbors in the same level: $s'_0, s'_1, s'_2, s'_3, s'_4, s'_5$.

be expressed in the form

$$\log p_{y,x^{(n)}|x^{(n+1)}}(y, x^{(n)}|\hat{x}^{(n+1)}) = \sum_{s \in S^{(n)}} l_s^{(n)}(x_s^{(n)}) + \log \tilde{p}_{x_s^{(n)}|x_{\partial s}^{(n+1)}, x_{\partial s}^{(n)}}(x_s^{(n)}|\hat{x}_{\partial s}^{(n+1)}, \hat{x}_{\partial s}^{(n)}) \quad (21)$$

which results in the following formula for the SMAP estimate of $X^{(n)}$:

$$\begin{aligned} \hat{x}_s^{(L)} &= \arg \max_{1 \leq k \leq M} l_s^{(L)}(k) \\ \hat{x}_s^{(n)} &= \arg \max_{1 \leq k \leq M} \{l_s^{(n)}(k) + \log \tilde{p}_{x_s^{(n)}|x_{\partial s}^{(n+1)}, x_{\partial s}^{(n)}}(k|\hat{x}_{\partial s}^{(n+1)}, \hat{x}_{\partial s}^{(n)})\} \\ &= \arg \max_{1 \leq k \leq M} \{l_s^{(n)}(k) - \log Z - \theta_1^{(n)} \sum_{r \in d(s)} \delta_{m \neq m_r} - \theta_2^{(n)} \sum_{q \in d^0(s)} \delta_{m \neq m_q}\} \end{aligned} \quad (22)$$

6 Estimation of Transition Probability Parameter

We introduced three parameters $\theta^{(n)} = [\theta_0^{(n)}, \theta_1^{(n)}, \theta_2^{(n)}]$ to determine the transition probability density function based on two different models of pyramid structure. To be specific, $\theta_0^{(n)}$ are required for fine-to-coarse in computing the log likelihood functions (17). And the $\theta_1^{(n)}, \theta_2^{(n)}$ are required for coarse-to-fine operations in the SMAP segmentation (22). The parameters are assigned with some initial value and can be updated in the process of segmentation by

solving a joint optimization problem specified by (17) and (22). This problem can be solved using ICM [3].

7 Estimation of GMM parameters

The statistical behavior of each segment is modeled by a Gaussian mixture distribution (GMM) with parameter $\eta_k = [\pi_k, \mu_k, R_k]$, $k = 1, \dots, M$, where M is the possible number of classes. Let $\pi_k \in R^N$, $\mu_k \in R^N$, $R_k \in R^{N \times N}$, where N is the number of Gaussian in the mixture density. And $\sum_{i=0}^{N-1} \pi_{k_i} = 1$, where π_{k_i} is the i^{th} entry of π_k . Then the probability density function of one pixel given its label k can be written explicitly as

$$p_{y_s|x_s,\eta}(y_s|k, \eta_k) = \sum_{i=0}^{N-1} \frac{\pi_{k_i}}{(2\pi)^{N/2}} |R_{k_i}|^{-1/2} \exp \left\{ -\frac{1}{2} (y_s - \mu_{k_i})^T R_{k_i}^{-1} (y_s - \mu_{k_i}) \right\} \quad (23)$$

For simpler case when $N = 1$, (23) can be reduced to

$$p_{y_s|x_s,\eta}(y_s|k, \eta_k) = \frac{1}{(2\pi)^{1/2}} |R_k|^{-1/2} \exp \left\{ -\frac{1}{2} (y_s - \mu_k)^T R_k^{-1} (y_s - \mu_k) \right\} \quad (24)$$

(23) and (24) are used to compute the log likelihood function defined in (16). We start the segmentation with some initially value of η_k . After applying the 3D SMAP algorithm to the entire data set Y , we obtain the label of pixels X .

Now we can divide the Y into M disjoint subsets. For each subset, we can use [2] to estimate new η'_k , with which we can apply the 3D SMAP algorithm again to refine the segmentation. This process is repeated until the change of GMM parameters between two successive iterations is less than certain amount of value. For instance, we can stop the iteration when the following criteria is satisfied.

$$\sum_{k=1}^M \|\mu'_k - \mu_k\|_{\infty} \leq \epsilon_3 \quad (25)$$

where $\|\cdot\|_{\infty}$ is the infinity norm and $\epsilon_3 = 1$.

The complete SMAP segmentation algorithm with parameter estimation is summarized below.

1. Set the initial parameter values for all n , $\theta_0^{(n)} = 5^{1-n}$, $\theta_1^{(n)} = 4$, and $\theta_2^{(n)} = 2$, $n = 0, \dots, L$.
2. Compute the likelihood functions using (17).
3. Compute $\hat{x}_s^{(L)}(k)$ using (22).
4. For scale $n = L - 1$ to $n = 0$
 - (a) Set $I =$ desired number of iterations
 - i. Compute $\hat{x}^{(n)}$ using (22).
 - ii. Update $\hat{\theta}^{(n)}$.
 - iii. Reduce I by one.
 - (b) Proceed if I is negative or no further change happens, otherwise go back to (a)
5. Re-estimate GMM parameters
6. Repeat steps 2 through 5 until (25) is satisfied.

8 Experiment Result

Data set: NiCrAl 1 keV Simulation, source from BlueQuartz Software.

Data Description: The data set consists of 257 slices of 513x513 pixel TIFF images. Each image represents a 15 nm thick slice, and the pixel dimensions are also 15x15 nm. Microscope parameter is 1 keV.

Experiment condition: The algorithm is implemented in C. The experiment for this particular dataset is performed on a PC with 2.0GHz clock speed, 1.5GB RAM. The running time of one iteration is 115.0 seconds including step 1 to 4 described above, importing and exporting images.

Segmentation results of the 201st to 205th image are provided as follows. The three different results are obtained by applying 2D SMAP, 3D SMAP without GMM parameters re-estimation and 3D SMAP with one iteration of GMM parameters re-estimation.

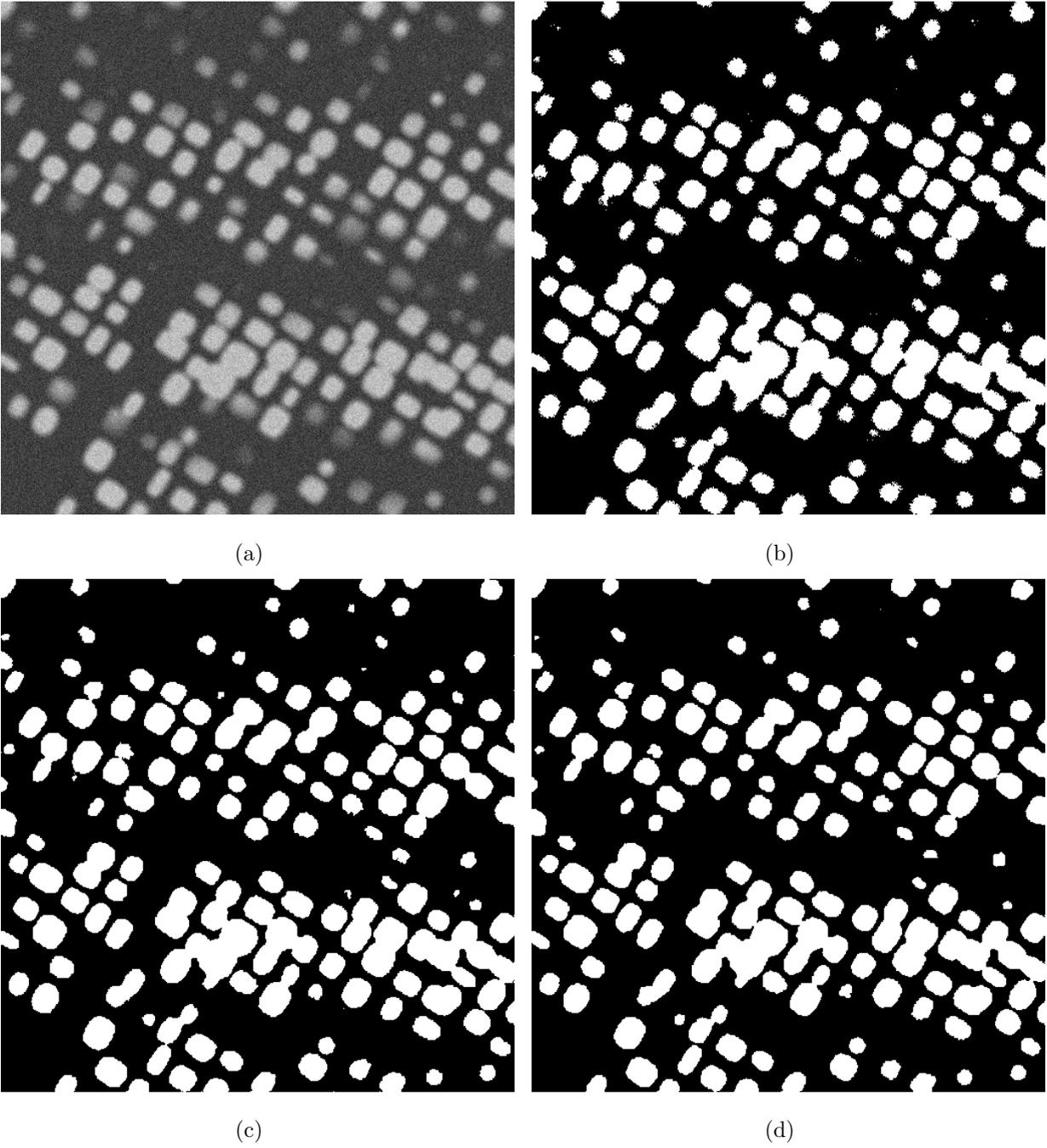


Figure 5: 201(a)Original image; (b)SMAP; (c)3D SMAP; (d)3D SMAP using re- estimated GMM parameters with number of iterations equals one

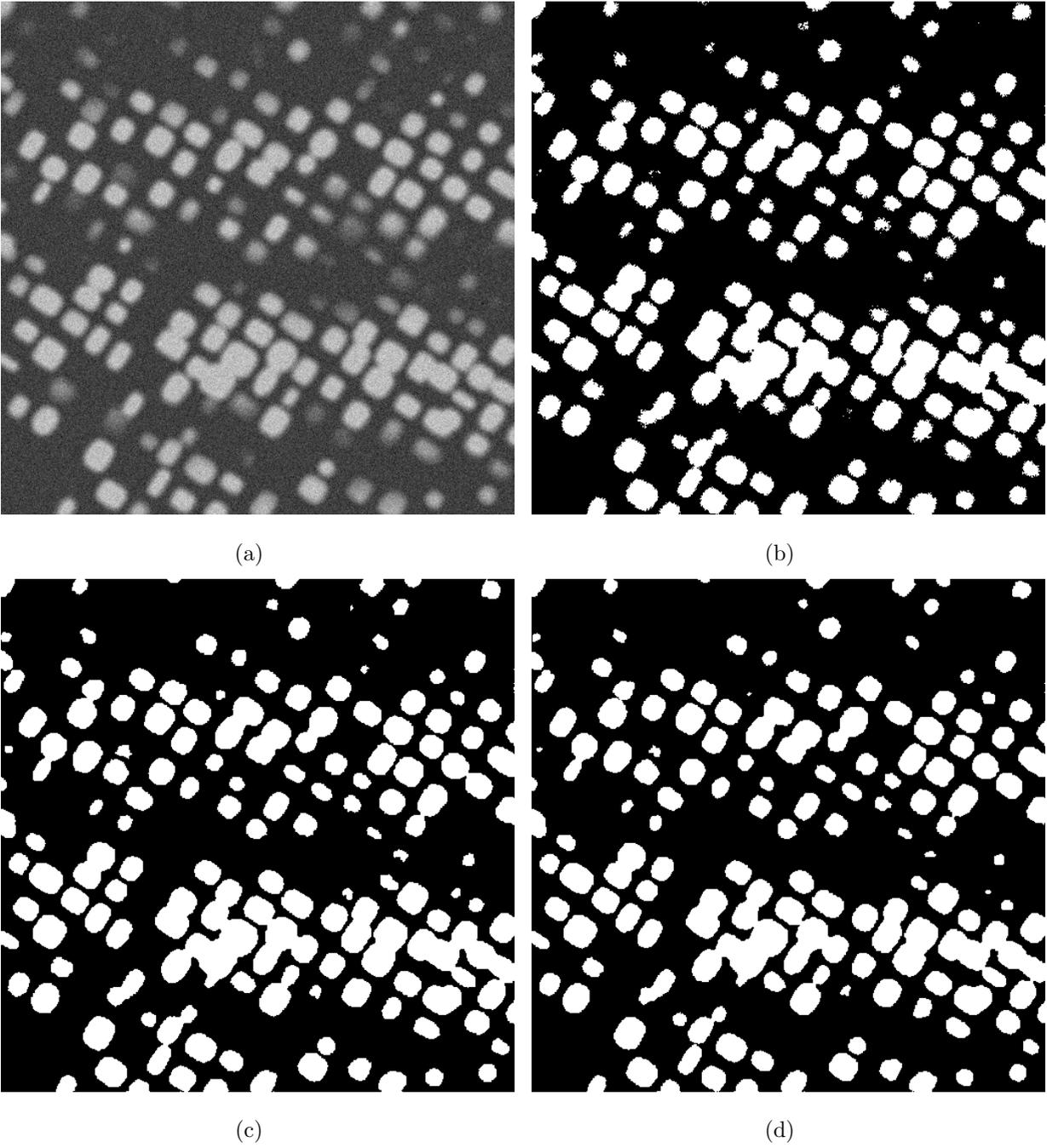


Figure 6: 202(a)Original image; (b)SMAP; (c)3D SMAP; (d)3D SMAP using re- estimated GMM parameters with number of iterations equals one

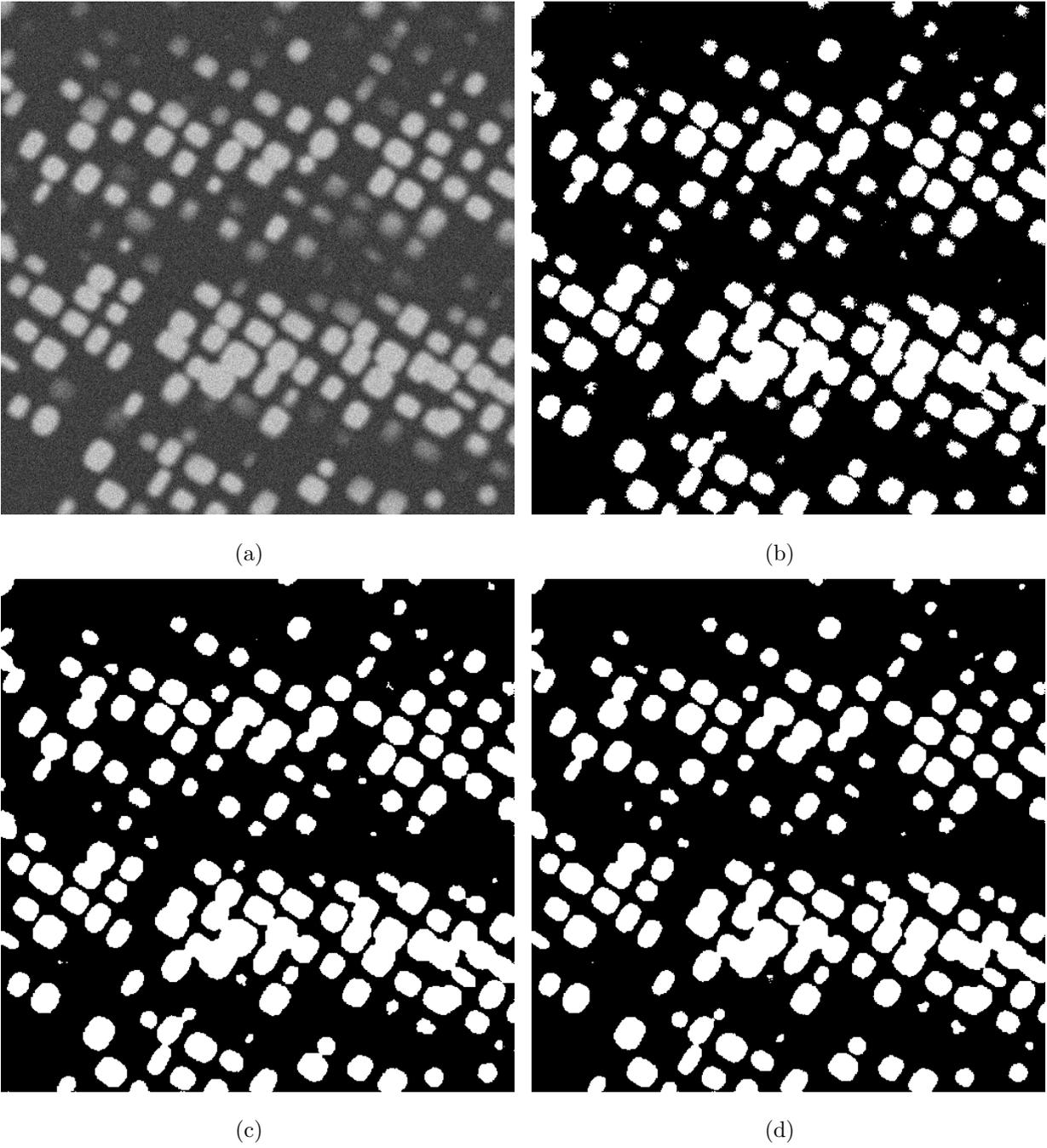


Figure 7: 203(a)Original image; (b)SMAP; (c)3D SMAP; (d)3D SMAP using re- estimated GMM parameters with number of iterations equals one

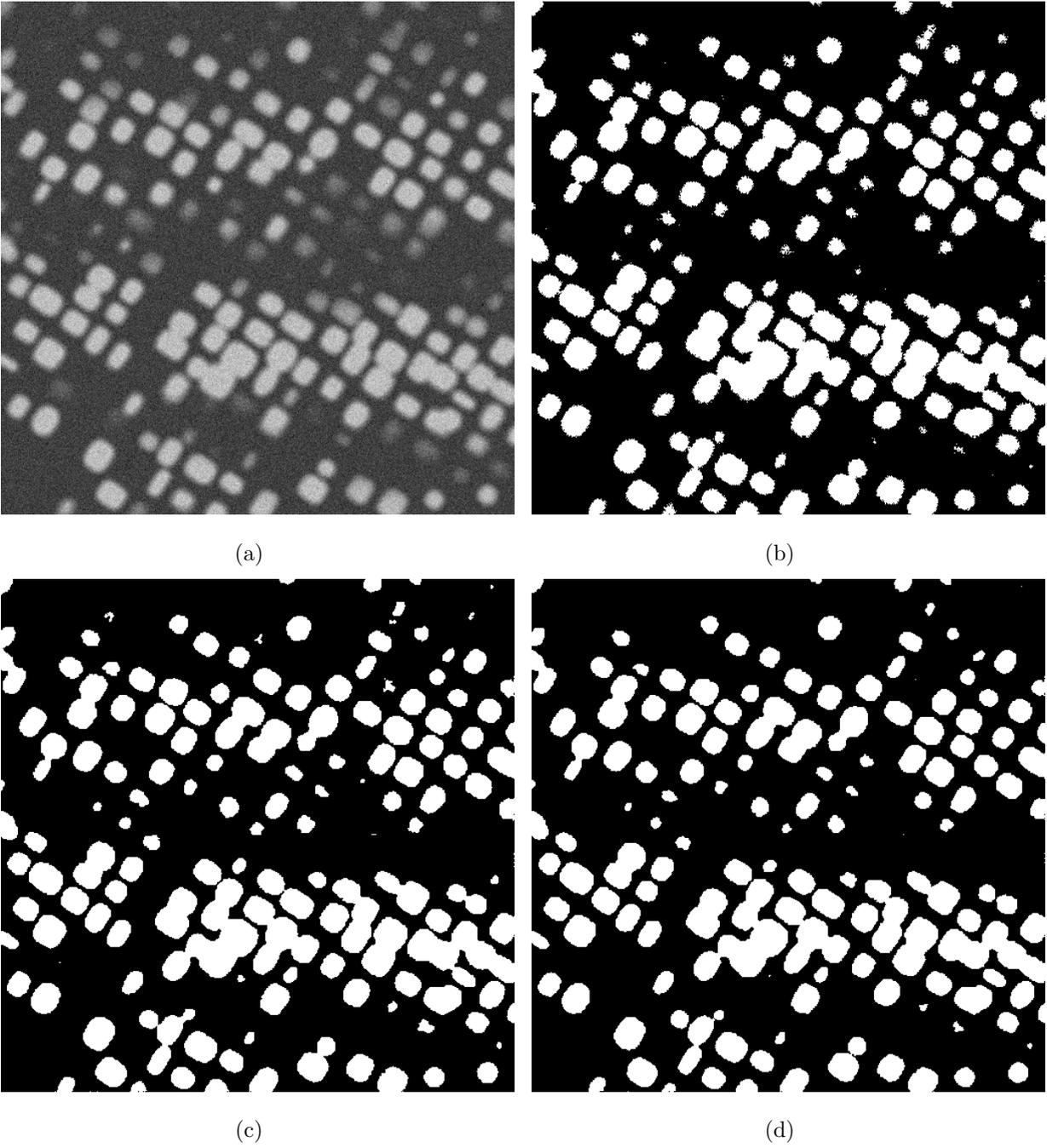


Figure 8: 204(a)Original image; (b)SMAP; (c)3D SMAP; (d)3D SMAP using re- estimated GMM parameters with number of iterations equals one

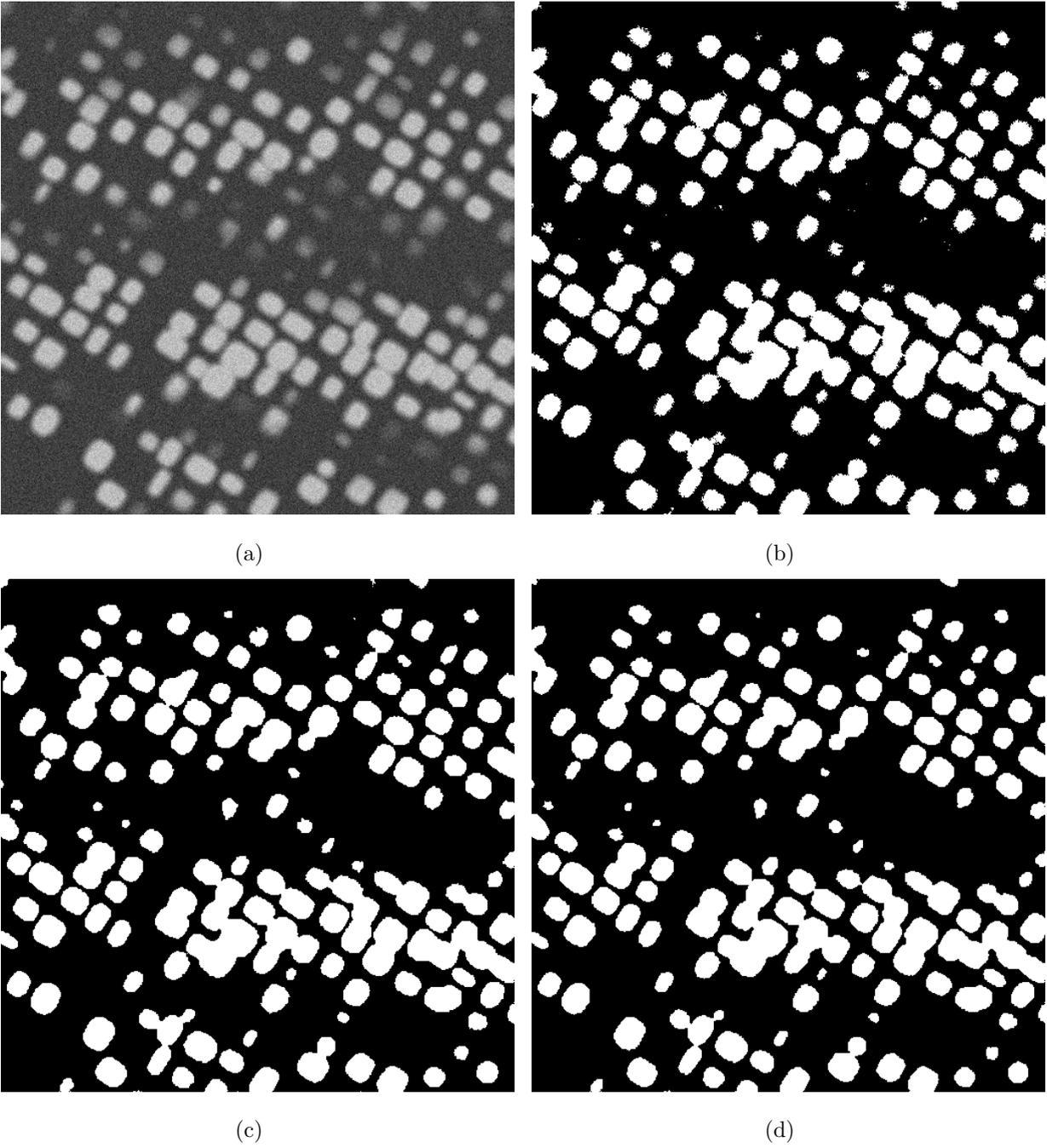


Figure 9: 205(a)Original image; (b)SMAP; (c)3D SMAP; (d)3D SMAP using re- estimated GMM parameters with number of iterations equals one

9 reference

- [1] C. A. Bouman and M. Shapiro, A multiscale random field model for Bayesian image segmentation, *IEEE Trans. Image Processing* 3:162-177, 1994.
- [2] C. A. Bouman, M. Shapiro, G. W. Cook, C. B. Atkins, H. Cheng, "Cluster: An unsupervised algorithm for modeling gaussian mixtures," 1998.
In <http://dynamo.ecn.purdue.edu/bouman/software/cluster>
- [3] C. A. Bouman, B. Liu, " Multiple Resolution Segmentation of Textured Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 2, pp. 99-113, Feb. 1991.