2006－CVIM－153 (22)
2006／3／16

自然画像合成のための準教師付き物体切り出し

杜 偉薇† 浦浜 喜一†

† 九州大学 大学院芸術工学研究院 視覚情報部門
福岡市南区塩原 4-9-1
E-mail: †duweiwei@gsd.design.kyushu-u.ac.jp, ††urahama@design.kyushu-u.ac.jp

あらまし 自然画像から物体を切り出して他の画像に貼り付ける画像合成では trimap と呼ばれるユーザの補助入力が利用される．本論文では，類似度データからファジィクラスタを抽出する準教師付き手法を応用して，少量の補助入力だけに基づいて物体領域を切り出す方法を提案する．本方法では精密な trimap ではなく，物体領域あるいは背景領域の 1 部をラフに指示入力するだけでよい．本方法でのメンバシップ伝搬のウインドウは広いので物体領域のギャップも飛び越えられ，少ない補助入力ですむ．ウインドウが広いと計算量が多くなるが，変数分離による高速近似アルゴリズムも示す．

キーワード 自然画像合成，物体切り出し，準教師付きクラスタ抽出，メンバシップ伝搬

Semi-Supervised Object Extraction for Natural Image Matting

Weiwei DU† and Kiichi URAHAMA†

† Faculty of Design, Kyushu University
Fukuoka-shi, 815-8540 Japan
E-mail: †duweiwei@gsd.design.kyushu-u.ac.jp, ††urahama@design.kyushu-u.ac.jp

Abstract Supporting strokes drawn by users called trimaps are exploited in natural image matting where an object is extracted from an image and composited against another image. In this paper, we apply a semi-supervised method for extracting a fuzzy cluster from similarity data to this task and present a method for extracting object regions by using coarse strokes by users. Rough specification of foreground or background regions instead of precise trimaps is sufficient for our method. Broad propagation window enables our method to jump gaps in objects and reduce specification strokes. However, broad window increases computational costs, hence we devise a fast algorithm for computing approximate solutions in the method.

Key words natural image matting, object extraction, semi-supervised cluster extraction, membership propagation

1. Introduction

Image matting refers to the process extracting an object from an image and compositing it against another image. Natural image matting instead of conventional blue or green screen matting has recently attracted attention, which is defined as

[Natural Image Matting] Given the color C of every pixel in a natural image, compute $x \in [0,1]$ and $C_F \in [0,255]^3, C_B \in [0,255]^3$ satisfying $C = xC_F + (1-x)C_B$ where suffix F refers to foreground and B is background.

This is a highly under-constrained problem difficult to solve at once. In ordinary approaches, x is computed at first and C_F and C_B are next computed. The variable x is called the alpha channel in computer graphics, while we call it the membership in this paper.

Computation of x is a fuzzy segmentation of the image into a foreground (object) region and a background area, which is rephrased as

[Object Extraction] Extract a fuzzy cluster with membership x from the set of pixels in an input image.

This is a problem still difficult to solve automatically. User interaction is hence normally exploited. Users are required to specify both of definitely foreground ($x = 1$) and definitely background ($x = 0$) pixels. The task of the matting system is then computation of x at remaining unknown pixels. Drawing such specification is called a trimap, an example of which is shown in Fig.1 where white is foreground, black is background and gray is unknown area. Narrower the unknown region, easier and more precise the system can estimate x there. However careful drawing of trimaps such as in Fig.1 is time-consuming and laborious. Moreover it is not uncommon to be hard to paint every object region in an image such as in Fig.6. Coarse drawing of only one region (normally background) is favorable for unskilled users.

Nevertheless many approaches to image matting require the trimap. Early approaches using sampling-based estimation need precisely drawn trimaps [1]~[3]. More modern approaches based on spatial propagation of memberships accept loose drawing but still require trimaps [4]~[6].

In this paper, we present a novel method for extracting objects from an image. Our method is based on the semi-supervised extraction of fuzzy clusters with arbitrary shapes from similarity data developed previously by us [7]. Our method can extract objects by using user's specification of only one region if the object is moderately easy to extract. Additional specification of another region, i.e. trimap drawing, improves the discrimination of objects hard to extract. In contrast to previous approaches where the spatial propagation of memberships is local, typically only through 4-neighbors, the window in our method is wide hence the propagation can jump across the gaps in objects. This gap surmounting capability of the propagation in our method can further reduce the labor of user's drawing.

2. Semi-Supervised Extraction of Fuzzy Cluster

Before presenting the matting method, we briefly overview a semi-supervised fuzzy cluster extraction method presented by us [7].

Let there be n data where the similarity between data i and j is $s_{ij} (= s_{ji}, s_{ii} = 0)$. Extraction of a cluster to which some data are known to belong is expressed as

$$\max \quad \sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij} x_i x_j - \sum_{i=1}^{n} f_i x_i^2 \tag{1}$$

where $f_i = \max\{\sum_j s_{ij}, \epsilon\}$ and $x_i \in [0,1]$ is the membership of datum i in the cluster. Eq.(1) resembles the Hopfield neural networks. Its first term produces a driving force for x_i increasing to 1, while the second term induces a counter-force to 0. Their balance gives the solution of eq.(1).

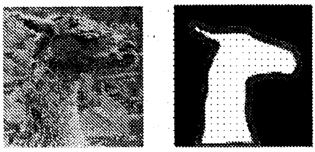


Fig. 1 Example of trimap (left: input image, right: trimap)

Some x_i's of specified data are fixed to 1 and the remaining x_i's of unknown data are computed with the fixed-point iteration:

$$x_i^{(\xi+1)} = \frac{1}{f_i}\sum_{j=1}^{n} s_{ij} x_j^{(\xi)} \tag{2}$$

where ξ is the iteration counter. All x_i is initially set to 0 except for fixed ones.

3. Object Extraction from Image

Let there be a color image with size $M \times N$ where the color of pixel (i,j) is C_{ij} which is represented with the CIELAB color space: $C_{ij} = (L*_{ij}, a*_{ij}, b*_{ij})$. The similarity between pixel (i,j) and $(i+k, j+l)$ is expressed by $s_{kl} = e^{-\alpha(k^2+l^2)-\beta\|C_{ij}-C_{i+k,j+l}\|^2}$ where s_{kl} is the abbreviation of $s_{ij,\{i+k,j+l\}}$. We set $s_{00} = 0$. For saving computation, we set $s_{kl} = 0$ if $|k| > p$ or $|l| > p$, i.e. we use the square window $-p \leqq k \leqq p, -p \leqq l \leqq p$.

Eq.(1) is , in this case, written by

$$\max \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=-p}^{p}\sum_{l=-p}^{p} s_{kl} x_{ij} x_{i+k.j+l} - \sum_{i=1}^{M}\sum_{j=1}^{N} f_{ij} x_{ij}^2 \tag{3}$$

where $f_{ij} = \max\{\sum_{k=-p}^{p}\sum_{l=-p}^{p} s_{kl}, \epsilon\}$ and eq.(2) becomes

$$x_{ij}^{(\xi+1)} = \frac{1}{f_{ij}}\sum_{k=-p}^{p}\sum_{l=-p}^{p} s_{kl} x_{i+k,j+l}^{(\xi)} \tag{4}$$

which resembles the bilateral filter [8].

Transformation of eq.(3) into an equivalent form of regularized interpolation is described in the appendix 1.

A user specifies some definitely foreground regions or definitely background regions by drawing strokes at those regions. The system then fix $x_{ij} = 1$ at pixels in those regions and computes x_{ij} at the remaining pixels by iterating eq.(3) until its convergence. If the foreground regions are specified by the user, computed x_{ij} is the membership of pixel (i,j) in the object. Conversely if the background regions are specified, $1 - x_{ij}$ is the degree of objects.

Owing to the cluster extraction capability of the method in section 2 with only specification of some cluster members, the present method can extract object with specification of only one of foreground or background regions. This is an advantage of the present method over the previous ones which require the specification of both foreground and background regions, i.e. drawing of trimaps.

Additional advantage in our method is the broad window in eq.(3) in contrast to only local, typically 4-neighbors, propagation in the previous methods. Broad window enables the propagation to jump gaps in objects as will be shown in the experiments below. Distant propagation between pixels simulates long-range connections between neurons in the visual area in brains. Broad window, however, increases the computational costs. Fast algorithm for computing eq.(4) with approximate spatial decomposition of s_{kl} is described in the appendix 2. This fast algorithm was used in all experiments below. Although this fast algorithm is only approximate, we can get with this algorithm results almost identical to those computed with exact computation of eq.(4).

If the color distributions are sufficiently different in foreground regions and background areas, then their discrimination is easy and we can extract objects with user's support of specification of only one of foreground or background regions. However if their distribution becomes close, their discrimination with the user's specification of only one of them becomes difficult. In such hard images, additional input of strokes of another regions serves to improve the extraction of objects. In such trimap specification cases, x_i's are fixed to 1 at the pixels specified to the foreground and are fixed to 0 at the background regions, and we compute x_i's at remaining unknown pixels.

4. Experiments of Object Extraction

We experimented firstly with the image of a parrot in Fig.2.

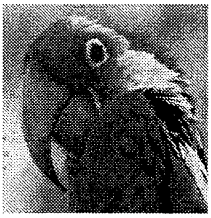
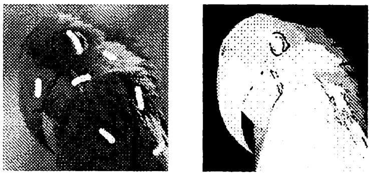
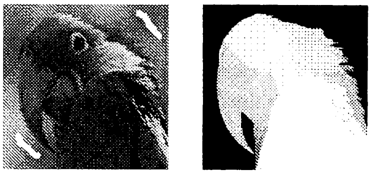


Fig. 2 Image of parrot



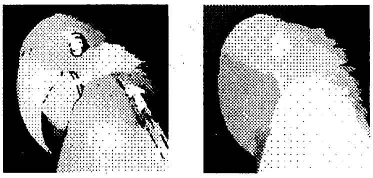
(a) strokes (b) memberships

Fig. 3 Extraction with specification of foreground



(a) strokes (b) memberships

Fig. 4 Extraction with specification of background



(a) with strokes in Fig.3(a) (b) with strokes in Fig.4(a)

Fig. 5 Memberships extracted with $p = 5$

This is an example where the object (parrot) is easy to extract with supporting strokes in either foreground or background. However easiness depends on which region is specified. Since the object (parrot) contains several color regions each of which requires touching of strokes, several strokes are needed for specifying foreground as is shown in Fig.3, while the background color is almost uniform hense few strokes are sufficient for specifying it as is shown in Fig.4. Parameters are $\alpha = 0.01, \beta = 0.1, \epsilon = 0.1, p = 50$ throughout this paper except for p which is varied in some figures. Note that filling up the periphery of the parrot in Fig.3 and the background space under the beak in Fig.4 is owing to the broad window in our method. If we set p small, these areas are not enough filled as is shown in Fig.5, hence additional strokes are needed if p is small. This space-filling capability of our method is also examined in the next example.

We experimented next with the image of a flower of spider lily in Fig.6. This is an example where the shape of the object (flower) is complex. We drew some strokes in the

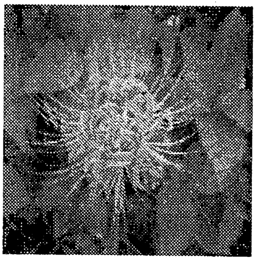


Fig. 6 Image of spider lily

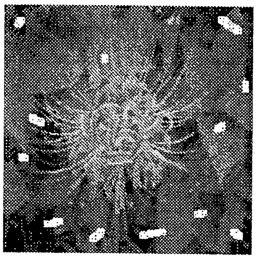


Fig. 7 Strokes in background in Fig.6

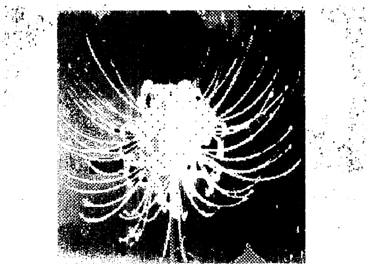


Fig. 8 Extracted memberships with $p = 10$

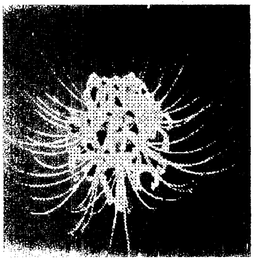


Fig. 9 Extracted memberships with $p = 50$

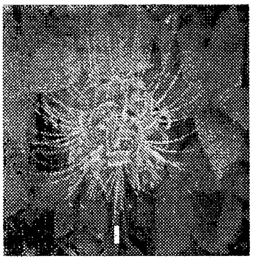


Fig. 10 Stroke (center bottom) in foreground in Fig.6

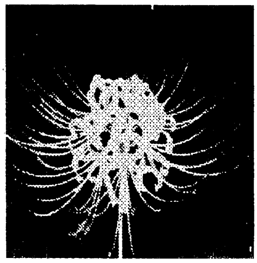
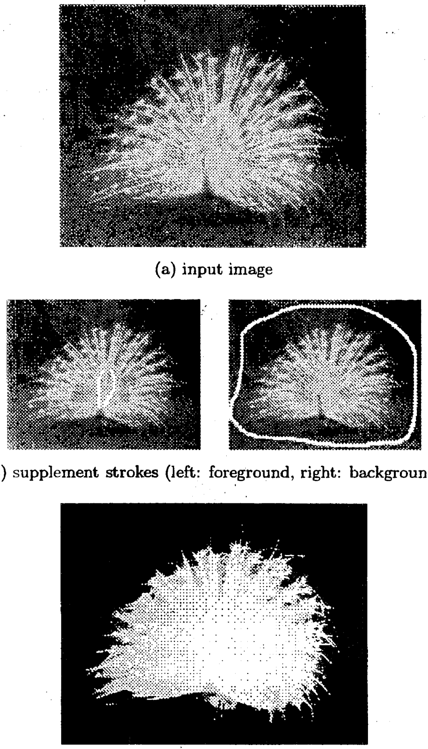


Fig. 11 Memberships extracted by using Fig.7 and Fig.10



(a) input image

(b) supplement strokes (left: foreground, right: background)

(c) extracted memberships

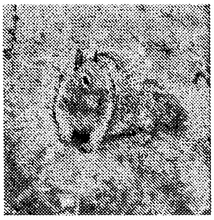
Fig. 12 Image of peacock

background region as shown in Fig.7. When we set $p = 5$, x_{ij} shown in Fig.8 was obtained. Some background regions isolated within the flower petals are failed to be discriminated as background and erroneously extracted as included in the object (flower). This is due to $p = 5$ is too narrow for x_{ij} to propagate across the petal. As far as we use narrow windows, we should draw many strokes, that is, at least one stroke in each isolated region. This is the case of previous methods [4]~[6] where memberships propagates to only 4 or 8 neighbor pixels. However this requirement for strokes drawn by users is laborious.

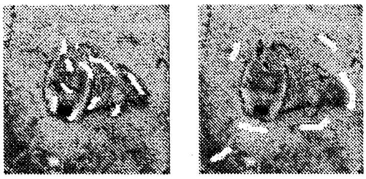
The result with $p = 50$ is shown in Fig.9 where the supplement strokes are the same as in Fig.7. Background regions enclosed in the flower are successfully extracted as backgrounds. This is owing to the broad window of $p = 50$ sufficiently wider than the width of the petals of the flower.

The extraction in Fig.9 is satisfactorily good but extraction of the stalk under the flower is weak because its color is close to the green of background leaves. Hence we added a stroke in the stalk as a foreground region shown in Fig.10. Then as shown in Fig.11, the stalk was also extracted clearly as objects. Since our method can be executed interactively, additional computation is fast because re-iteration of eq.(4) starts from the final values of x_{ij} of previous iterations. This suitability to interactive processing is a merit of iterative solution methods.

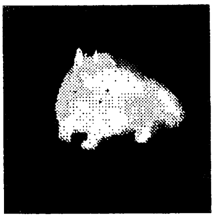
Next Fig.12 is also an example of object (peacock) with complex shape. We drew strokes both in foreground and background. Final example is an image of squirrel in Fig.13



(a) input image



(b) supplement strokes (left: foreground, right: background)



(c) extracted memberships

Fig. 13 Image of squirrel

where the object (squirrel) is hard to extract because its color is close to that of backgrounds.

5. Image Composition

As we described in the introduction, objects extracted from an image are composed with another image in image matting. This composition needs the foreground color C_F of every pixel in addition to their membership x. In previous approaches to image matting [1]~[6], both of C_F and C_B are computed. Their computation is , however, complex and wasteful because C_B is unnecessary for the image composition and is discarded. Hence in this paper, we present a new scheme for computing only the foreground color C_F by using the memberships x computed with the above algorithm and the color C of the input image.

Let the color of pixel $(i.j)$ in an input image be $C_{ij} = [R_{ij}, G_{ij}, B_{ij}]$ and its precomputed membership be x_{ij}. We compute the foreground color $C_{Fij} = [R_{Fij}, G_{Fij}, B_{Fij}]$ by

$$\min \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=-p}^{p}\sum_{l=-p}^{p} s_{kl}x_{i+k,j+l}\|C_{Fij}-C_{i+k,j+l}\|^2 \quad (5)$$

where s_{kl} is the same as that in eq.(3). Eq.(5) is solved analytically as

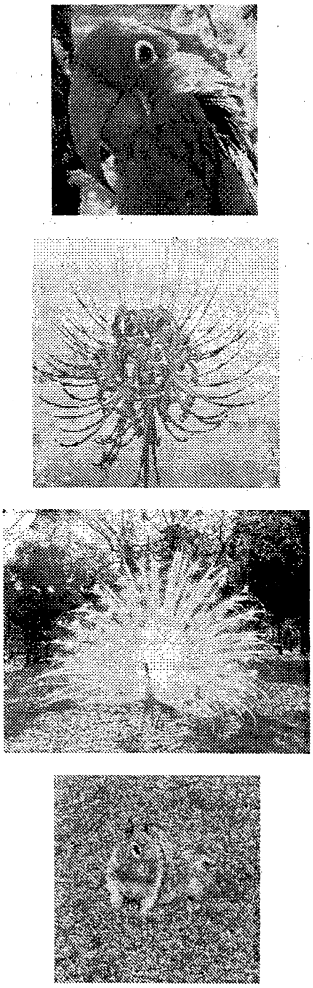


Fig. 14 Composite images with new backgrounds

$$C_{Fij} = \frac{\sum_{k=-p}^{p}\sum_{l=-p}^{p} s_{kl}x_{i+k,j+l}C_{i+k,j+l}}{\sum_{k=-p}^{p}\sum_{l=-p}^{p} s_{kl}x_{i+k,j+l}} \quad (6)$$

where x_{ij} is added to eq.(4) as a third weight, so eq.(6) is of the form of the trilateral filter [9]. Eq.(6) is also computed approximately with a fast algorithm similar to that for eq.(4) as is described in appendix 2.

Let the color of pixel (i, j) in another image with the same size $M \times N$ be D_{ij}. Then we set $x_{ij}C_{Fij} + (1 - x_{ij})D_{ij}$ to the color of pixel (i, j) in the composite image. Examples are shown in Fig.14.

6. Conclusion

We have presented a method for extracting objects from natural images for image matting. Examination of various color coordinates for their suitability to our method is currently under study.

Reference

[1] M. Ruzon and C. Tomasi, "Alpha estimation in natural images", Proc. CVPR, pp.18-25, 2000.

[2] Y. Y. Chuang, B. Curless, D. Salesin and R. Szeliski, "A Bayesian approach to digital matting", Proc. CVPR, pp.264-271, 2001

[3] S. Lin and J. Shi, "Fast natural image matting in perceptual color space", Computers & Graphics, 29, pp.403-414, 2005.

[4] L. Grady, T. Schiwietz and S. Aharon, "Random walks for interactive alpha-matting", Proc. VIIP, pp.423-429, 2005.

[5] V. Vezhnevets and V. Konouchine, "GrowCut: Interactive multi-label N-D image segmentation by cellular automata", Proc. Graphicon, 2005.

[6] J. Wang and M. C. Cohen, "An iterative optimization approach for unified image segmentation and matting", Proc. ICCV, pp.936-943, 2005.

[7] W. Du, K. Inoue and K. Urahama, "Unsupervised and semi-supervised extraction of fuzzy clusters in similarity data", IEICE Tech. Report, PRMU2005-177, pp.165-170, Jan. 2006.

[8] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images", Proc. ICCV, pp.839-846, 1998.

[9] P. Choudhury and J. Tumblin, "The trilateral filter for high contrast images and meshes", Proc. Eurographics Sym. rendering, pp.186-196, 2003.

Appendix

1. Equivalent Formulation of Eq.(3)

Eq.(3) is rewritten as

$$\min \quad \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=-p}^{p}\sum_{l=-p}^{p} s_{kl}(x_{ij} - x_{i+k,j+l})^2/2 + \sum_{i=1}^{M}\sum_{j=1}^{N} g_{ij}x_{ij}^2 \qquad \text{(A·1)}$$

where $g_{ij} = f_{ij} - \sum_{k=-p}^{p}\sum_{l=-p}^{p} s_{kl}$. This is the form of regularization which is interpreted as the Bayesian inference and implemented with an electronic circuit which is also related to random walks. If $\epsilon = 0$, then the second term in eq.(A.1) vanishes. Note again that the novelty of eq.(A.1) lies in the broad window in contrast to the local 4 or 8 neighbors interaction in Markov random fields or electrical circuits.

2. Fast Algorithms for Eq.(4) and Eq.(6)

The following algorithm is derived from an approximate decomposition of $e^{-\beta\|C_{ij}-C_{i+k,j+l}\|^2}$ as $e^{-\beta\|C_{ij}-C_{i+k,j}\|^2}\cdot e^{-\beta\|C_{i+k,j}-C_{i+k,j+l}\|^2}$.

[Algorithm for eq.(4)]:

Step 1: For all i and j, compute

$$y_{ij} = \sum_{l=-p}^{p} x_{i,j+l}^{(\xi)} e^{-\alpha l^2} e^{-\beta\|C_{ij}-C_{i,j+l}\|^2} \qquad \text{(A·2)}$$

$$g_{ij} = \sum_{l=-p}^{p} e^{-\alpha l^2} e^{-\beta\|C_{ij}-C_{i,j+l}\|^2} \qquad \text{(A·3)}$$

Step 2: For all i and j , compute

$$z_{ij} = \sum_{k=-p}^{p} y_{i+k,j} e^{-\alpha k^2} e^{-\beta\|C_{ij}-C_{i+k,j}\|^2} - x_{ij}^{(\xi)} \qquad \text{(A·4)}$$

$$h_{ij} = \sum_{k=-p}^{p} g_{i+k,j} e^{-\alpha k^2} e^{-\beta\|C_{ij}-C_{i+k,j}\|^2} \qquad \text{(A·5)}$$

$$u_{ij} = \max\{h_{ij} - 1, \epsilon\} \qquad \text{(A·6)}$$

$$x_{ij}^{(\xi+1)} = z_{ij}/u_{ij} \qquad \text{(A·7)}$$

This $x_{ij}^{(\xi+1)}$ is an approximation of eq.(4). Computational time of eq.(4) until convergence is shown in Fig.A.1.

[Algorithm for eq.(6)]:

Step 1: For all i and j, compute

$$y_{ij} = \sum_{l=-p}^{p} x_{i,j+l}C_{i,j+l} e^{-\alpha l^2} e^{-\beta\|C_{ij}-C_{i,j+l}\|^2} \qquad \text{(A·8)}$$

$$g_{ij} = \sum_{l=-p}^{p} x_{i,j+l} e^{-\alpha l^2} e^{-\beta\|C_{ij}-C_{i,j+l}\|^2} \qquad \text{(A·9)}$$

Step 2: For all i and j , compute

$$z_{ij} = \sum_{k=-p}^{p} y_{i+k,j} e^{-\alpha k^2} e^{-\beta\|C_{ij}-C_{i+k,j}\|^2} - x_{ij}C_{ij} \qquad \text{(A·10)}$$

$$h_{ij} = \sum_{k=-p}^{p} g_{i+k,j} e^{-\alpha k^2} e^{-\beta\|C_{ij}-C_{i+k,j}\|^2} \qquad \text{(A·11)}$$

$$u_{ij} = \max\{h_{ij} - x_{ij}, \epsilon\} \qquad \text{(A·12)}$$

$$C_{Fij} = z_{ij}/u_{ij} \qquad \text{(A·13)}$$

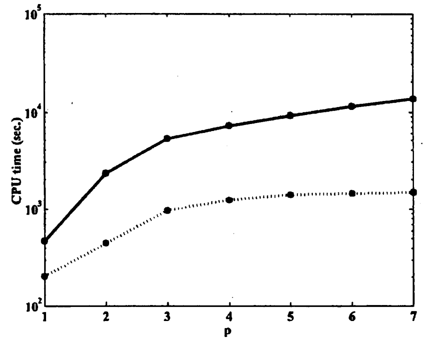


Fig. A·1 Computational time (solid line:eq.(4), dotted line: fast algorithm)