SEMI-AUTOMATIC PROBABILISTIC MORPHOLOGICAL DETECTION

Frans M. Coetzee

GenuOne, Inc.
2 Copley Square, Boston MA

Visvanathan Ramesh

Siemens Corporate Research
College Road East, Princeton NJ

ABSTRACT

We describe a semi-automated approach for designing morphological operators to detect image structures. We automatically combine morphological primitives from a pool to generate an enumerable family of complex morphological operators spanning a Receiver Operating Curve (ROC). Domain knowledge can be encoded by biasing the pool with primitives; the system subsequently automatically selects and combines operators based on the joint statistics of training data. The major advantages are that the designer can focus on constructing simple operators, yet is able to rapidly combine them to yield more powerful, system-specific solutions, whose operating point can easily be changed. We illustrate the approach using Birth and Death processes and associated operators. Examples of video text detection are presented.

1. INTRODUCTION

Reliable detection of pixels corresponding to man-made structures is of cardinal importance in computer vision. Examples of particular industry interest are billboards in broadcast video, inventory numbers on shipping containers, license plates on vehicles, and symbols on legacy CAD images.

While morphology yields excellent detection, these systems require complex, handcrafted design, and face major difficulties with controlling the operating point[1]. We present an approach to morphological pixel detection that addresses these problems. The approach reflects our experience with morphological operators: a designer usually has no problem producing many simple operators that each plausibly matches specific image shapes. Humans, however, fail at appropriately combining different operators, especially in noise; or given a system, weighing different operators to select the operating point.

In our approach the prior domain knowledge about basic shapes and marginal statistics can be encoded in the form of many partial morphological solutions, handcrafted or selected from a generic primitive library. However, complex interactions between primitives are automatically extracted from training data to optimally combine the operators. The design process is practical since it is deterministic and requires at most two passes through the training set. It yields a set of solutions that define a Receiver Operating Curve, allowing for an operating point to be selected without difficulty. Finally, since the primitive operators are simple and generic, it becomes feasible to have a portable design approach suited to many applications.

We compare our approach to other semi-automatic morphological design methods. An extensive review of binary morphological design by Dougherty[7] is recommended. We distinguish three main groups. The formal methods define grammars suited to scene descriptions, and attempt to derive operators directly from a generative scene description. For example, Joo[6] generates the morphological solution using a predicate calculus solver. These approaches hold out the attraction of generating provably optimal operators with no intervention. Unfortunately, real vision effects such as occlusion, shadowing and noise are too complex for current approaches. These approaches fail to capture the probabilistic uncertainties of real applications, and usually cannot be parameterized using example images.

The second group depend on generic genetic and learning algorithms to search operator space e.g. [2, 3, 4]. The computational complexity is much higher than the method proposed here. These methods further force on the designer the non-transparent problem of encoding prior domain information in the form of genomic fitness functions, or neural architectures. Solutions cannot easily be modified to achieve a range of operating points (the approach in [5] offers only an approximate solution by combining the false alarm rate and probability of detection in a fitness function).

Our method falls into the third group. These decompose some operators, and use training data to select between operators. The major compromise we make is to restrict our tools for combining simple operators to be non-parametrized set operations (union, intersection and complement). The combinations are automatically constructed from a large representative training set. This restriction in practice limits our approach to detection and classification problems. Our method is less ambitious than the full formal approaches in the sense of not yielding full morphological operator decompositions. However, using a subset of the morphological algebra yields significant benefits: hand-designed morphological solutions can be integrated into the overall solution, yet operating points along the ROC can easily be changed, and the final solution can be presented in human understandable form as a Boolean expression. Our method does allow for the production of *non-increasing operators*. It is also robust since the design is based on a representative sample of images and noise, rather than a model.

In the remainder of the paper for simplicity we consider binary image morphology, where the image and the indicator set are identified. We note that we perform grayscale processing by quantizing the image using a Gray or level code, and processing multiple image bitplanes simultaneously.

2. MORPHOLOGICAL PROCESSING

Morphology is defined on a set E with closed addition operator $+ : E \times E \to E$. The core primitives are dilation $\oplus$ and erosion $\ominus$ of a subset A by a subset (structuring element) B:

$$A \oplus B = \{a + b | a \in A\} = \cup_{b \in B} A_b \quad (1)$$

$$A \ominus B = \{a | a + b \in A\} = \cap_{b \in B} A_{-b} \quad (2)$$

More complex operators are constructed by combining primitives using set operations ($\cap, \cup, ^c$). In image processing, A represents an input image, while the other sets are called structuring elements. The set operations of a morphological operator can be considered parameterized on a structuring element, or not. For example, the *hit-and-miss* operator on A by structuring elements J and K, is

$$A \otimes (J, K) \quad = \quad (A \ominus J) \cap (A^c \ominus K) \tag{3}$$

Here the component operators $(A \ominus J)$ and $(A^c \ominus K)$ are considered parameterized (henceforth called templates), and the connecting intersection operator non-parameterized (henceforth referred to as the *feature logic*. This distinction formalizes that typical morphological operator design involves matching simple structuring elements to desired features (templates), and deriving complete solutions by using Boolean logic to represent high-level constraints across these features.

While yielding excellent performance, morphological systems are notoriously hard to design, since the operators are not differentiable and practical systems require combining operators in long chains. For example, a designer can easily produce a template T_1 that matches letter positions on a license plate, while a second template T_2 matches the plate outline. Both templates can detect the license plate under ideal circumstances, but in noisy practical scenarios their performances differ markedly. For example, license plate boundaries frequently merge with the background under IR illumination, while the letters remain visible. To build a system using these templates, the designer can use any one of sixteen possible binary functions including the following:

$$T_1(x), T_2(x), T_1(x) \cap T_2(x), T_1(x) \cup T_2(x), T_1\bar{(}x) \cap T_2(x)$$

In practice, the optimal choice may depend on accuracy requirements imposed by law and differ between deployments.

3. DESIGN APPROACH

The design process is shown in Figure 1. Simple templates are used as a library. Test images are processed with each template, yielding a set of images where each pixel is the output of an template centered on that pixel in the input image. The large pool is pruned to maximize divergence between pixels in ROI and pixels not in each ROI, as discussed below. The remaining templates are combined using Neyman-Pearson design to yield a parameterized set of Boolean expressions $(B_1, B_2, B_3..)$ of the simple templates, and which span the ROC. To change the operating point, the closest Boolean combination is used.

3.1. Pool Pruning and Feature Logic Design

In practice, the number of templates that can be combined by any classifier training procedure is limited. Using simple generic operators, vast numbers of templates can result which even video data cannot parameterize without overfitting. We therefore perform a two-stage pass through the data, first pruning the template pool, then optimizing the combination.

The discrimination value of templates can be estimated only when considered in pairs or larger combinations. Such feature selection for larger combinations is extremely expensive (cf.[8]). In this application we use a simple pairwise procedure based on the K-L divergence being a rough proxy of the area under the ROC curve. Consider first pairs of templates indexed by i. The symmetric Kullback-Leibler distance of each feature pair i from the rest of the feature pairs $j, j \neq i$, averaged over all classes, is the selection criterion. For every unique template pair (v_{i_1}, v_{i_2}) $i_1, i_2 = 1, 2, \ldots N, i_1 \neq i_2$, a histogram $p_i^c[k]$ on the training data for every target class $c, c = 1, 2, \ldots M$ is generated. The exclusion probability distribution $p_i^{c*}[k]$ for every class c for each feature pair i is first calculated

$$p_i^{c*}[k] \quad = \quad \left[\sum_{j \neq c}^{{}^N C_2} p_i^j[k] \mathcal{P}(j)\right] / \left[\sum_{j \neq c}^{{}^N C_2} \mathcal{P}(j)\right] \tag{4}$$

The average distance between the per-feature true and false class problem across all classes is then calculated:

$$D_i^* = \sum_{c,k} \left[p_i^c[k] \log(p_i^c[k]/p_i^{c*}[k]) + p_i^{c*}[k] \log(p_i^c[k]/p_i^{c*}[k])\right] \tag{5}$$

The template pairs are ranked based on this divergence, and as many of the top ranked templates as allowed by system resources, are selected for integration.

The next step requires optimal combination of morphological template outputs, and automatically providing Boolean expressions for different achievable operating points.

Each template in the pool produces a classification mapping from the image X to the space $\{0, 1\}$ at each image pixel. The feature logic design should solve the problem of taking these N templates $f_i : X_i \rightarrow \{0, 1\}$, to find the classifier $F : \prod_{j=1}^N f_i(X_i) \rightarrow \{0, 1\}$ produced by combining the outputs of the templates that optimizes detection at a given false alarm rate. In general, testing all 2^{2^N} distinct different combination rules $F_j, j = 1, 2, \ldots 2^{2^N}$ for combining the simple templates is out of the question (for $N = 5$, there are more than $2^{2^N} \simeq 4.3 \times 10^9$ functions). Taking into account the probabilistic nature of the input, the problem is solved by the Neyman-Pearson lemma[9]. Under hypotheses H_0 and H_1 on the input space χ, we have induced distributions $\mathcal{P}(y_i|H_0)$ and $\mathcal{P}(y_i|H_1)$ on the feature space of bit-packed template outputs $y = f(\chi)$. The ROC curve is spanned by 2^N classifiers, each of which is associated with one of the finite set of thresholds in the set

$$Q \quad = \quad \{\zeta_i | \zeta_i = \mathcal{P}(H_1|y_i)/\mathcal{P}(H_0|y_i), \;\; y \in f(\chi)\} \tag{6}$$

where the thresholds are strictly ordered, so that $\zeta_i < \zeta_{i+1}, i = 1, 2, \ldots ||Q|| - 1$. Our implementation of the above procedure is as follows: we have a significant amount of labeled training data, corresponding to pixels in regions of interest. We can reliably extract accurate histograms for templates at ROI pixels and their complements from the training images. The binary outputs of the templates at each pixel are packed into a bit-vector that is used directly as the binary representation of the integer index of a histogram bin. The thresholds ζ are calculated using ratios on the histograms in each bin, and sorted. For a given ROC operating point (a selected value of ζ), the decision regions are generated by labeling bins with the index of ζ, since each value of ζ has a one-to-one association with classification confidence. A Boolean representation of the binary decision rule is directly extracted from the bin indices and simplified using efficient standard automatic Boolean logic provers such as the Quinn-McCluskey method. We store the classifiers on the ROC, and, to change operating point, switch between these.

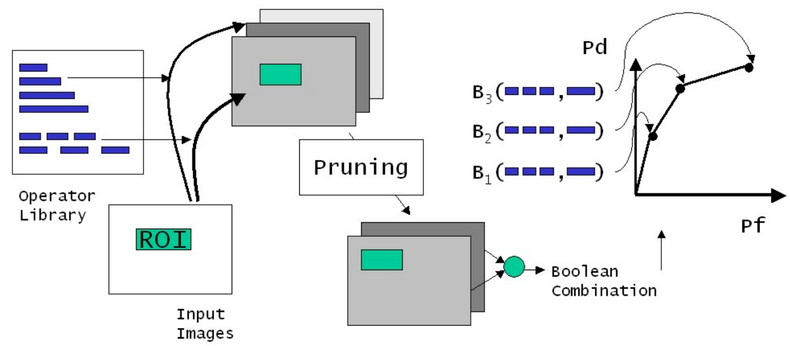


Fig. 1. Summary of design process. Simple templates, some generic, and some hand-coded by designers, form a library. Test images are processed with each template. The large pool of templates is pruned to maximize divergence between pixels in ROI and pixels not in each ROI. The subset of templates are combined using a Boolean procedure to yield a parameterized set of operators that span an ROC.

3.2. Generic Template Design

In principle, given unlimited computing power and data, we can include every plausible structuring element in the original template library. In practice, we find good performance for our surveillance applications by using templates matched to coupled birth-and-death processes. These are multi state systems where each state transition produces random samples corresponding to interval lengths as in Figure 2. For scenes imaging relatively few agents, the model is a reasonable description of the scan-lines at different orientations. The template pool then consists of simple openings and closings using line elements of different angles, sizes and spacing.

A full analysis of the statistical behavior of these processes under morphology is beyond the space limitations of the current paper. We use a simple example for motivation: it shows that morphological primitive operations matched to such sequences can encode complicated joint statistics into marginal distributions of the output sequences, which can then easily be separated by a second series of basic templates. Hence, higher order statistics can be detected and exploited by the operator combinations our approach provides.

Consider the process shown in Figure 2. It produces events consisting of three positive pulses of fixed size α_0 separated by two notches of possible sizes β_0 and β_1. The joint distribution on the notch configuration is $p_{ij} = P(\beta_i, \beta_j),\ i, j = 0, 1$. Figure 2 shows as the effect of closing the sequences with operators of size γ_1, where $\beta_0 \leq \gamma_1 < \beta_1$ and γ_2, where $\beta_1 \leq \gamma_2$.

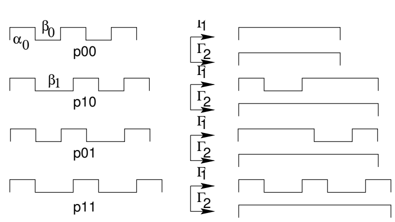


Fig. 2. Events produced by a two-state process where pulses of constant width α_0 are separated by two notches drawn from two values, β_0 and β_1, after closing by operators of widths $\beta_0 < \gamma_1 \leq \beta_1$ and $\beta_1 \leq \gamma_2$. The probability of a notch sequence (β_i, β_j) is indicated by p_{ij}.

Figure 3 shows the marginal distributions $p(\alpha|_\gamma)$ for the original, as well as the closed sequences. It is clear that distinctive

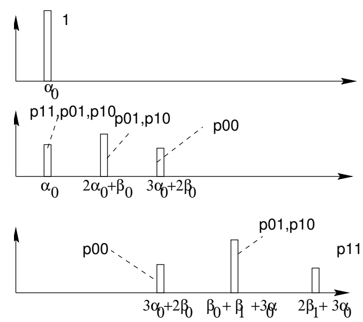


Fig. 3. Encoding of joint statistics of β in the marginal distribution of α after closing. (a) $p(\alpha)$, (b) $p(\alpha|_{\gamma_1})$, (c) $p(\alpha|_{\gamma_2})$. In each case the dependence of the marginal distribution value on the joint probability p_{ij} of the two notch sequence is shown.

marginal distributions appear in the regions between critical values of γ. In particular, modes appear depending on the joint distribution of the notches. Since pulses deriving from a certain mode can be detected via successive openings, it is possible to detect events from a specific birth-death process based on both marginal and joint statistics of β, by performing a sequence of closings, and a subsequent set of openings. Similarly, separation on the basis of the statistics of α can be performed using openings and subsequent closings. Specific forms of invariance result from choices of templates; in this example it is not possible to separate processes differing only in p_{01} and p_{10}; as expected, since the library has only symmetric templates (a line element). Using different spacings and line elements, however, a form of frequency analysis and a-symmetric processing is readily performed.

4. APPLICATIONS

The system has been tested on a number of industrial text detection tasks, where it replaced commercial OCR systems[10]. The latter systems, optimized to document handling, perform poorly on data characterized by high-speed, low resolution video, and surveillance camera angles.

Our first application is license plates detection for vehicle control, which remains a challenge due to legal rulings worldwide that require accurate detection under all weather conditions. Warping is applied to remove perspective distortion, allowing for one set of detectors to be applied uniformly across the image. Equalization and prefiltering is performed and grayscale images are mapped to multiple bitplanes. The bitplanes are processed in parallel and outputs are combined across all planes and all templates. We pruned the template pool from eighty elements and ultimately integrated twenty templates, trained on 1000 video sequences.

The result on a typical image is shown in Figure 4. The top image shows an original pre-filtered image at one wavelength before quantization. The middle image shows each pixel color coded according to the likelihood ratio between the two classes (the threshold ζ_i). We found this visualization of confidence level to be very useful as a design tool. The classifier found the most reliable features to use to be the spaces, and combination of spaces between the letters of the license plate, in contrast to our initial assumption that the rectangular shape of the license plate would be the critical feature. Figure 4(c) shows the per-pixel ROC curve achieved.

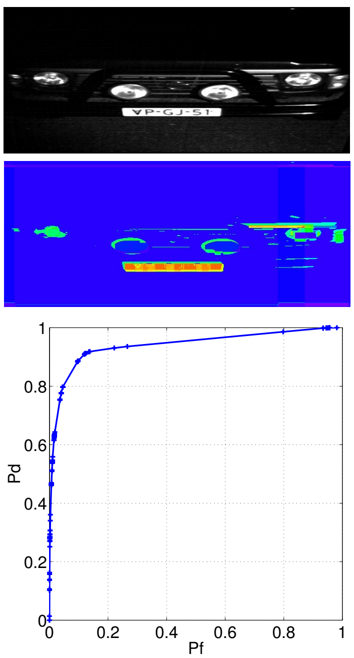
(a)
(b)
(c)


Fig. 4. License plate detection. (a) Original image, after warping, (b) posterior likelihood ratio, (c) ROC curve for individual license plate pixels.

Figure 5 shows the results on a container identification problem, which is subject to significant lighting, clutter, and occlusion problems. The challenge here is that the scaffolding moves across the image and contains significant frequency overlap with the text. Figure 5(a) shows a typical binarized image bitplane, while Figure 5(b) shows the confidence levels for different text lines. In this case, the algorithm once again favors different combinations of inter-character spacings, but also favors operators matching large blocks of letter hits.

5. CONCLUSION

We presented a method for designing morphological detectors. The approach uses a generic classifier to combine the outputs from a large number of simple templates from a design pool. The approach avoids the need for the designer to struggle to combine multiple classifiers, and allows for statistically setting the operating point based on image data.

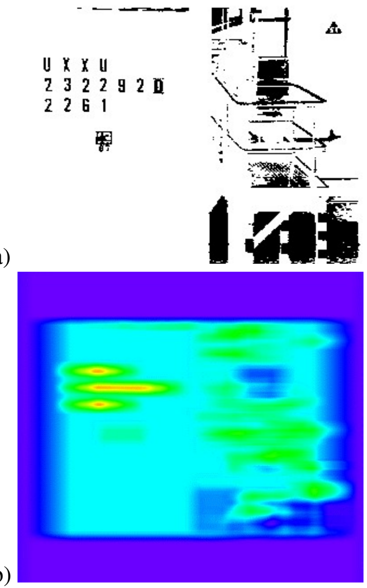
(a)
(b)


Fig. 5. Container text detection. (a) Binarized original image, (b) final confidence of text line centroid after grouping.

6. REFERENCES

[1] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Transactions PAMI*, vol. 9, no. 4, pp. 532–550, 1987.

[2] Neal R. Harvey and Stephen Marshall, "The use of genetic algorithms in morphological filter design," *Signal Processsing: Image Communication*, vol. 8, pp. 55–71, 1996.

[3] Ikushi Yoda, Kazuhiko Yamamoto, and Hiromitsu Yamada, "An automatic acquisition of hierarchical mathematical morphology procedures by GA," *Image and Vision Computing*, 1999.

[4] Micahel A. Zmuda, Louis A. Tamburino, and Mateen M. Rizki, "An evolutionary learning system for synthesizing complex morphological filters," *IEEE Trans. Systems, Man and Cybernetics (B)*, vol. 26, no. 4, pp. 645–653, Aug 1996.

[5] Michael E. Bullock, David L. Wang, and Tim J. Patterson, "Automated training of 3-D morphology algorithm for object recognitions," in *Proc. SPIE ATR*, 1994, vol. 2234.

[6] Hyonam Joo, *Automatic Morphology*, Ph.D. thesis, University of Washington, November 1991.

[7] E. R. Dougherty, "Automatic design of morphological operators," *Journal of Electronic Imaging*, vol. 13(3), pp. 486–491, 2004.

[8] F.M. Coetzee, S. Lawrence, and C. L. Giles, "Bayesian classification and feature selection from finite data sets," in *Proceedings UAI2000*, Stanford, CA, July 2000, UAI.

[9] H. L. Van Trees, *Detection Estimation and Modulation Theory*, vol. 1-3, Wiley and Sons, 1971.

[10] S. V. Rice, F. R. Jenkins, and T. A. Nartker, "The fifth annual test of OCR accuracy," Tech. Rep. TR96-01, Information Science Research Institute, University of Nevada - Las Vegas, April 1996.