

Supervised Segmentation by Iterated Contextual Pixel Classification

Marco Loog and Bram van Ginneken

Image Sciences Institute

University Medical Center Utrecht

HP E.01.335, P.O. Box 85500, 3508 GA Utrecht, The Netherlands

{marco,bram}@isi.uu.nl

Abstract

We propose a general iterative contextual pixel classifier for supervised image segmentation. The iterative procedure is statistically well-founded, and can be considered a variation on the iterated conditional modes (ICM) of Besag. Having an initial segmentation, the algorithm iteratively updates it by reclassifying every pixel, based on the original features and, additionally, contextual information. This contextual information consists of the class labels of pixels in the neighborhood of the pixel to be reclassified. Three essential differences with the original ICM are: 1) Our update step is merely based on a classification result, hence avoiding the explicit calculation of conditional probabilities; 2) The clique formalism of the Markov random field framework is not required; 3) No assumption is made w.r.t. the conditional independence of the observed pixel values given the segmented image. The important consequence of Properties 1 and 2 is that one can easily incorporate common pattern recognition tools in our segmentation algorithm. Examples are different classifiers—e.g. Fisher linear discriminant, nearest-neighbor classifier, or support vector machines—and dimension reduction techniques like LDA, or PCA. We experimentally compare a specific instance of our general method to pixel classification, using simulated data and chest radiographs, and show that the former outperforms the latter.

1. Introduction

Segmentation is a crucial step in image analysis. Pixel-based methods are one of the most important and often applied approaches. In addition to direct pixel classification methods, there are also iterative pixel-based approaches. The latter mainly fall within the field of probabilistic relaxation labelling, often using a Markov random field framework [8], or compatibility coefficients and support functions [5] to formulate the segmentation task. These ap-

proaches model the spatial relationships between pixels in the original image and the segmented image within a certain neighborhood statistically. By incorporating such neighborhood information, they try to improve upon segmenting images by direct pixel classification.

This work proposes an iterative approach, that we call iterated contextual pixel classification (ICPC). An initial segmentation for the ICPC is obtained by a pixel classification based on one or several features. These features can be anything, as long as they are merely constructed from information that is directly available from the image. Typical examples are the gray value of a pixel or certain kinds of filter outputs, the location of the pixel, etc. Given this current segmentation, i.e., a class labelling for all pixels, we can reclassify all pixels based on the foregoing features and additional features extracted from the current segmentation. We call these latter features *contextual*. Examples are the labels of pixels within a certain neighborhood of the pixel to be reclassified. The reclassification of all pixels provides us with a new segmentation, and we can repeat the reclassification procedure. (See for Sect. 2).

ICPC is similar to the iterated conditional modes (ICM) of Besag [1] (see also [2], [4], and [8]). However, there are three essential differences between our method and the original formulation of the ICM, which we discuss in Sect. 4. In addition, these differences allow us to plug in established pattern recognition techniques into ICPC, such as different classifiers, ranging from neural networks to support vector machines, or dimension reduction techniques like linear discriminant analysis, self-organizing maps, or principal component analysis.

The paper is organized as follows. Sect. 2 introduces the basic notations, formalizes the segmentation problem and the statistical theory behind our model. Moreover, it shows how to formulate the update step by a simple classification task, and it discusses our experimental setup. Sect. 3 describes the experimental results on artificial and medical data. Sect. 4 provides conclusions and a discussion.

2. Iterated contextual pixel classification

2.1. The preliminaries

Let $x = (x_1, \dots, x_n)$ be a vector of n feature vectors. Every x_i is associated with a pixel p_i in the image L . So every x_i consists of, for example, a single feature like the gray value in this pixel, or of multiple features like the gray value and a certain number of filter outputs associated with this pixel. Of course, many other features are possible, such as the location in the image, etc.

Let $s = (s_1, \dots, s_n)$ be a vector of the same length, where every s_i contains the label that belongs to this pixel. Labels are in the set $K = \{1, 2, \dots, k\}$, where k is finite. Pixels having the same label are in the same segment of the image. A vector s is said to be a segmentation of L .

The problem of finding an optimal segmentation s of L , can be defined statistically. We consider x and s to be realizations of a stochastic process and consider the probability $P(s|x)$, being the probability of having s as a segmentation of the image L , given the features x . The optimal segmentation s^* is the one that maximizes the probability $P(s|x)$:

$$s^* = \operatorname{argmax}_{s \in K^n} P(s|x). \quad (1)$$

2.2. The basic update step

The ICPC provides an approximation to s^* by iteratively updating the segment labelling of single pixels. However, before we come to the ICPC, we first consider the general idea behind the update step to find an approximate solution to the optimization problem in (1).

If s and ζ are both segmentations of L , only differing from each other in one entry, say, the i th, then using Bayes' theorem, we find that

$$\frac{P(s|x)}{P(\zeta|x)} = \frac{P(s, x)}{P(\zeta, x)} = \frac{P(s_i | s_{-i}, x)}{P(\zeta_i | \zeta_{-i}, x)} = \frac{P(s_i | \zeta_{-i}, x)}{P(\zeta_i | \zeta_{-i}, x)}, \quad (2)$$

where ζ_{-i} stands for $(\zeta_1, \dots, \zeta_{i-1}, \zeta_{i+1}, \dots, \zeta_n)$.

Assuming that all probabilities $P(s|x)$ are larger than zero, we have from Eq. (2), that $P(s|x) > P(\zeta|x)$ if and only if

$$P(s_i | \zeta_{-i}, x) > P(\zeta_i | \zeta_{-i}, x). \quad (3)$$

Hence, in the case that $P(s|x) > P(\zeta|x)$, accepting segmentation s in favor of ζ , would give us a segmentation that is closer to the optimal s^* , and we see that the decision, whether s is better or worse than ζ , can be based solely on the comparison of the conditional probabilities on the single i th entry. Therefore, iterating this procedure of changing one of the labels of the segmentation ζ and accepting the new segmentation s , if and only if inequality (3) holds, leads to an iterative approach which finds an approximate solution to Eq. (1) (cf. [2] and [8]).

2.3. The ICPC update step

In going from segmentation ζ to s , we could randomly choose a new class label for entry i , determine $P(s_i | \zeta_{-i}, x)$ and $P(\zeta_i | \zeta_{-i}, x)$, and decide whether to update the segmentation or not, i.e., accept or reject s , respectively. However, what we choose to do is to pick that class label for s_i that maximizes the conditional probability $P(s_i | \zeta_{-i}, x)$ right away. Hence, in the update step of the ICPC, we always decide to have s as our new segmentation, because we always have that $P(s|x) \geq P(\zeta|x)$.

Interestingly, this update step can be seen as a classification task in which the vector (ζ_{-i}, x) is to be classified to one of the k classes in K . Bayesian classification suggests that (ζ_{-i}, x) should be classified to that class c for which $P(s_i = c | \zeta_{-i}, x)$ is largest, which is precisely the update step we suggest in the ICPC—in practice, we assume a Markov property [2, 3, 8] to hold and we will not use the complete context ζ_{-i} (cf. SubSect. 2.5).

2.4. The general ICPC algorithm

In the description of the general setup, C denotes a non-contextual pixel classifier, which maps a feature vector x_i , as described in Subsect. 2.1, to one of the possible classes in K . C^+ is a contextual classifier, which maps (c_i, x_i) , a vector containing both the contextual feature vector c_i and the non-contextual feature vector x_i , to a class in K . With these definitions we can write down the general ICPC steps.

- 1) Construct x_i for every pixel from the image L .
- 2) Classify every x_i by means of C , to obtain the initial segmentation, i.e., set s_i equal to $C(x_i)$.
- 3) Start a cycle in which every pixel is visited exactly once and carry out the following steps.
 - a) Pick an s_i from s , according to some visiting scheme.
 - b) Construct the vector c_i from s not including s_i .
 - c) Combine c_i and x_i to one feature vector.
 - d) Reclassify s_i , i.e., set $s_i = C^+(c_i, x_i)$.
- 4) Repeat step 3 until no pixels are relabelled during a complete cycle.

2.5. Features, classifiers, and more

One of the key points of the ICPC is its general formulation: We do not have to specify, what kind of gray value features, contextual feature, classifier, etc. we use. However, before we can do actual experiments, we of course have to fill in these 'unknowns'.

In our experimental setup we use a first nearest neighbor (1NN) classifier [6] both in the non-contextual and contextual case. In the training and testing phase the feature vectors are first linearly transformed such that the average within-class covariance matrix equals the identity matrix,

i.e., the within-class scatter is pre-whitened. The contextual feature vector is build up of class labels of pixels within a certain radius ρ from the pixel to be reclassified, e.g, if ρ is set to one, we take the four labels associated to the four direct neighbors of this pixel, if it is set to two, we take twelve neighboring labels into account, etc. The initial feature vectors x_i consist of the gray value of the pixel itself and additional gray values of pixels within a certain radius r not necessarily equal to ρ . Furthermore, in the experiments on medical data we also include two position features, which are the pixel's x and y coordinates in the image.

Further, the visiting scheme mentioned in 3a can be stochastic or deterministic as long as we visit every pixel exactly once during one cycle. In our setup we use a stochastic scheme: In every step we pick uniformly an entry that has not yet been visited during this cycle.

One final remark: The class labels of the s_i , of which we build up the contextual feature vector are categorical, rather then numerical. The INN classifier can only handle numerical feature vectors. We solve this by determining k vectors in a $k - 1$ dimensional space in such a way that the Euclidean distance between all vectors is equal—e.g., if $k = 4$, the vectors span an equilateral tetrahedron in 3 dimensions. Every vector represents one of the classes, and we take these vectors as part of our contextual feature vector—e.g., if we want to include the class labelling of the four closest neighbors of a pixel in our feature vector, and the number of possible classes is four, this would mean that $3 \cdot 4 = 12$ contextual entries are included.

3. Experimental results

3.1. Artificial images

The first experimental results we present are on artificial images. We constructed¹ thirty 125×125 images with 5 classes, including background. We generated 10 test and 20 train examples. Two images, together with their ground truth, are shown in the top four sub-figures in Fig. 1 (the originals are on the left).

We did a segmentation by pixel classification (PC) for 4 different radii of the feature neighborhood. For the ICPC we used the same features and fixed the contextual neighborhood to 4, i.e, $\rho = 1.0$. The ICPC converged within 20 iterations per image. The pixel classification errors (PCEs) of the experiments are in Table 1. We see that ICPC clearly outperforms PC consistently. In Fig. 1 (bottom), we show the segmentations of the second example image, obtain by the best ICPC (right), and best PC (left) settings.

We also tried ICPC with contextual neighborhood radii of 1.5 and 2.0, but, although they did improve upon PC,

¹The images used in this test as well as the Mathematica notebook that generated them can be obtained from the first author.

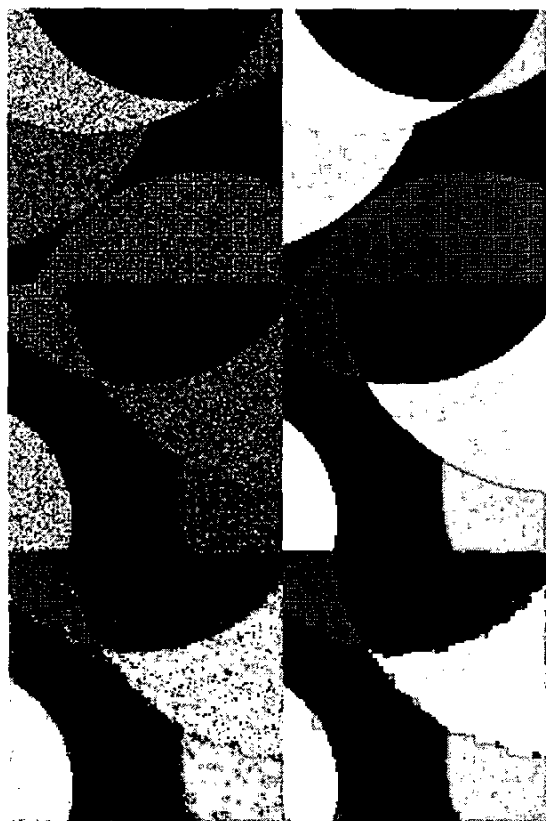


Figure 1. Illustrations to Subject 3.1.

there was no improvement w.r.t. the ICPC with $\rho = 1.0$.

3.2. Lung field segmentation

The second results are on standard chest radiographs, which come from the database described in [7]. The task is to segment the two lung fields. We distinguish between 3 classes: left lung, right lung, and background. The images are 256 by 256 pixels. 50 are used for training, 20 for testing. In Fig. 2, we show examples of—starting in the upper-left corner and going clockwise—a chest radiograph, its ground truth, its ICPC segmentation, its PC segmentation. The PCE results are in Table 2. We see that ICPC performs much better than PC. Furthermore, the PCE of 3.1%

Table 1. PCEs in % on artificial images.

r	1.0	1.5	2.0	2.5
PC	25	18	12	9
ICPC	14	8	5	4



Figure 2. Illustrations to Subject 3.2.

Table 2. PCEs in % on chest images.

r	0.0	1.0	1.5	2.0
PC	9.0	5.2	6.2	6.3
ICPC	7.8	3.1	3.3	3.4

obtained by ICPC with $r = \rho = 1.0$, comes very close to performance of the best automatic scheme discussed in [7]. This latter scheme is hybrid and combines rule-based segmentation and pixel classification.

4. Discussion and conclusions

We introduced a general approach to segmentation, called iterated contextual pixel classification (ICPC), in which we iteratively reclassify pixels based on contextual and non-contextual information. The specific ICPC instance, based on a INN classifier, outperformed non-iterative pixel classification in our experiments on artificial, and chest-radiograph images.

The approach is similar to Besag's ICM [1]. However, three essential differences between our method and the original formulation of the ICM should be mentioned—along with them we make some additional comments.

1) The update step, in which for every pixel it is determined to which class it belongs, is merely based on a simple classification result. Hence, we avoid the explicit calculation of conditional probabilities.

In our experiments, of course, we could have determined this conditional probability by means of the INN classifier

if necessary for the segmentation task it is not.

2) To model the statistics present in a certain pixel context, we do not require the clique formalism [2, 3, 8] as used in the Markov random field (MRF) framework in which the original ICM was formulated. This makes it easier to deal with large contexts and complex interactions in our approach, because we do not need to determine and model all cliques present in a certain neighborhood. Within the MRF theory, pixel interactions are often restricted to small neighborhoods or pair-wise pixel interactions (cf. [3]).

Another drawback of most MRFs is that the statistical model is usually assumed to be parametric, most often even normal. Our approach, on the contrary, can be used in a fully non-parametric setting (see also the next property).

3) The assumption, given the segmented image, that the observed pixels in our original image are independently distributed is dropped. This has been suggested before in [4], for example. However, as opposed to these two contributions, our approach is not restricted to a local auto-normal formulation for the conditional distribution. Generally, we can even use fully nonparametric models like nearest neighbor or kernel estimators, or, especially in the 2-class case, support vector machines.

In addition, Properties 1 and 2 allow us to plug in established pattern recognition techniques into ICPC. For example, classifiers ranging from neural networks to support vector machines, or dimension reduction techniques like linear discriminant analysis, self-organizing maps, or principal component analysis. The use of dimension reduction techniques might enable ICPC to cope with high-order interactions and large contextual and non-contextual neighborhoods.

References

- [1] J. Besag. Discussion of paper by P. Switzer. *Bull. Int. Statist. Inst.*, 50(3):422–425, 1983.
- [2] J. Besag. On the statistical analysis of dirty pictures. *J. Roy. Stat. Soc. B*, 48(3):259–279, 1986.
- [3] G. L. Gimel'farb. *Image Textures and Gibbs Random Fields*. Kluwer Academic Publishers, Dordrecht, 1999.
- [4] J. Kittler and J. Föglein. Contextual decision rules for objects in lattice configuration. In *Proc. 7th ICPR*, pages 270–272, Montreal, 1984.
- [5] J. Kittler and J. Illingworth. Relaxation labelling algorithms—a review. *Im. Vis. Comput.*, 3(4):206–216, 1985.
- [6] D. Mount and S. Arya. ANN: Library for approximate nearest neighbor searching. <ftp.cs.umd.edu/pub/faculty/mount/ann/>.
- [7] B. van Ginneken and B. M. ter Haar Romeny. Automatic segmentation of lung fields in chest radiographs. *Med. Phys.*, 27(10):2445–2455, 2000.
- [8] G. Winkler. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*. Springer-Verlag, Berlin, 1995.