2005

TR-2005001: Automatic Target Detection in E3D Images using Mathematical Morphology Techniques

Ilknur Icke

Jose Hanchi

Robert M. Haralick

Automatic Target Detection in E3D Images using Mathematical Morphology Techniques

Ilknur Icke, Jose Hanchi and Robert M. Haralick
The Graduate Center, City University of New York
Pattern Recognition Laboratory / Room 4-321
365 Fifth Avenue; New York, NY 10016
email:{iicke,hanchi,haralick}@gc.cuny.edu

Abstract

In this paper, we present an Automatic Target Detection system that operates on a simulated E3D(Exploitation of 3D Data) image dataset. Simulated E3D images are range images where each value represents the height above the ground. In our work, we treat the 3D data as if the height values were pixel intensity values(2D) and a set of mathematical morphology operations are applied to each image to generate features and then a decision tree classification algorithm is used to distinguish between the target and non-target pixels in test images. The experimental results show over 99% *target detection performance.*

1. Introduction

The goal of our work is to efficiently and accurately detect the target objects in simulated E3D images. In a simulated E3D image, each pixel value represents the height above the ground. Each image may contain different types of objects with different shapes.We define the problem as a classification problem in which we have a large number of E3D images with some objects marked as the targets while everything else is marked as non-target.

The first step is to generate features that would let us distinguish the target objects from non-target objects.Each feature represents the result of a mathematical morphology operation on a given pixel. Since the morphological operators relate to the shape characteristics of the objects, they give better results than using convolution based operators [4].We were also concerned with generating a large number of features efficiently.Finally we trained a binary decision tree on these features as our classifier to make the target/non-target decision on line.

This paper is organized as follows: Section 2 provides the necessary background on gray scale mathematical morphology. In section 3, we describe the feature generation process. The decision tree classification algorithm along with the accuracy results on two separate datasets are given in section 4. Finally, section 5 details the feature elimination algorithm and the experimental results we obtained.

2. Gray Scale Mathematical Morphology

In the following section, we will give the definitions of four basic gray scale mathematical morphology operations as described in [4].

2.1. Definitions

Let $F, K \subseteq E^{N-1}$ and $f : F \rightarrow E$ and $k : K \rightarrow E$ where E is the Euclidean Space.

Dilation $f \oplus k : F \oplus K \rightarrow E$ can be computed as:

$$(f \oplus k)(x) = \underset{\substack{z \in K \\ x-z \in F}}{max} \quad \{f(x-z) + k(z)\}$$

Erosion $f \ominus k : F \ominus K \rightarrow E$ can be computed as:

$$(f \ominus k)(x) = \underset{z \in K}{min} \quad \{f(x+z) - k(z)\}$$

Opening $f \circ k : F \circ K \rightarrow E$ can be computed as:

$$(f \ominus k) \oplus k$$

Closing $f \bullet k : F \bullet K \rightarrow E$ can be computed as:

$$(f \oplus k) \ominus k$$

2.2. Efficient Implementation of Gray Scale Mathematical Morphology Operations

Given a *zero-height flat structuring element* k, i.e. $k(x) = 0$ for all x, dilation can be computed as

$$(f \oplus k)(x) = \underset{x \in F}{max} \quad \{f(x)\}$$

likewise, erosion can be computed as

$$(f \ominus k)(x) = \min_{x \in F} \quad \{f(x)\}$$

In this manner, dilation corresponds to a *max*-filter and erosion to a *min*-filter. There are various algorithms to calculate these filters. We implemented the Van Herk-Gil-Werman Algorithm [2] because of its efficiency. Opening and closing operations were implemented using a combination of dilation and erosion operations as in the definitions given in Section 2.1.

3. Feature Generation

Given an image I and a set of zero-height flat structuring elements $\mathcal{S} = \{S_1, S_2, ..., S_n\}$ we perform the morphology operations of (i) erosion, (ii) dilation (iii) opening and (iv) closing.

We used three kinds of structuring elements:

- H_s: a horizontal line of s
- V_s: a vertical line of size s
- B_s: a square box of side size s, constructed with a horizontal and vertical element of size s

The sizes of the structuring elements come from the set

$$\{3, 5, 7, 9, 11, 13, 15, 17, 19, 21\}$$

The morphology operations for the line structuring elements were computed using erosion and dilation as follows:

- Erosion: $I \ominus L_s$
- Dilation: $I \oplus L_s$
- Opening : $I \circ L_s = (I \ominus L_s) \oplus L_s$
- Closing : $I \bullet L_s = (I \oplus L_s) \ominus L_s$

The morphology operations for the box structuring elements are defined as a combination of horizontal and vertical line structuring elements:

- Erosion : $I \ominus B_s = (I \ominus H_s) \ominus V_s$
- Dilation: $I \oplus B_s = (I \oplus H_s) \oplus V_s$
- Opening : $I \circ B_s = (((I \ominus H_s) \ominus V_s) \oplus H_s) \oplus V_s$
- Closing : $I \bullet B_s = (((I \oplus H_s) \oplus V_s) \ominus H_s) \ominus V_s$

Since we have three types of structuring elements (H, V, B), each of ten sizes and four kinds of morphology operations, we effectively generate $3 \times 10 \times 4 = 120$ morphology features per pixel of every image. In other words, we map every pixel to a vector of 120 morphology features (Figure 1).

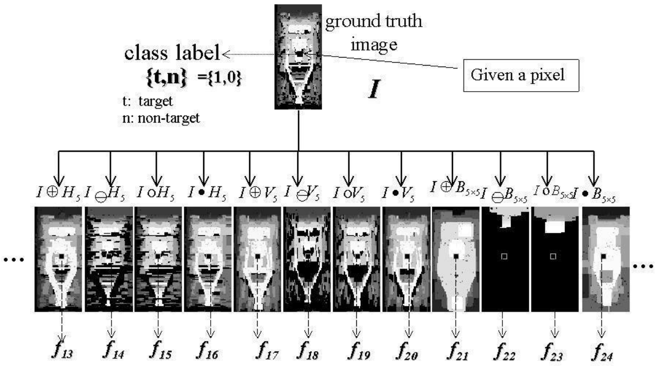

Figure 1: An image is presented to form a set of feature images

4. Classification Algorithm

We had two separate image datasets with two different resolution values. Our first dataset of *resolution 75mm* had 44 images of targets surrounded by non-target regions and 345 images containing non-target regions only. The second dataset of *resolution 200mm* had 34 images containing targets surrounded by non-target regions and 689 images containing mostly non-target regions along with some target pixels.

The task was to classify each pixel as either *target(1)* or *non-target(0)*. The classification algorithm we chose for this task was the *binary decision tree* algorithm using a *thresholding decision rule* [3]. This is a CART based algorithm [1] and our software along with some example datasets are available on our website (http://www.prl.gc.cuny.edu).

4.1. Decision Tree Training

We generated a decision tree T_d for each dataset, i.e. $T_{d|75mm}$ and $T_{d|200mm}$. The outline of the process we followed is:

1. For every pixel with coordinates (x, y) from an image I, i.e. $I(x, y)$, extract the vector of 120 morphology features

$$\mathcal{F}(I(x,y)) \rightarrow \underbrace{(I_{f_1}(x,y), \ldots, I_{f_{120}}(x,y))}_{\textbf{feature vector}}$$

2. Extract the ground truth label $L_{I(x,y)} \in \{0, 1\}$ for that pixel and create a *labeled* ground-truth vector

$$(I_{f_1}(x,y), \ldots, I_{f_{120}}(x,y), L_{I(x,y)})$$

The original image and coordinate information are not retained, i.e. no references to I, x, y are kept after this point.

3. Generate a training dataset D_{train} with 2/3 of all *target* vectors and a comparable amount of *non-target* vectors.
4. Generate a testing dataset D_{test} with the remaining 1/3 of *target* vectors and a similar amount of *non-target* vectors.
5. Train the decision tree T_d with D_{train} and measure the accuracy of the decision tree by comparing the result of applying the decision tree classifier to each vector of D_{test} based on the ground truth labels.

4.2. Classification Accuracy Results

The contingency tables given below show the classification accuracy results for the decision trees trained for each dataset. As it is stated in the previous section, the training and test datasets are randomly sampled from the feature datasets and they are independent of each other.

- **Results on the 75mm Dataset**

  Size of $D_{train|75mm}$: 292,831 feature vectors

  Size of $D_{test|75mm}$: 1,783,434 feature vectors

		Assigned Class	
		Non-target	Target
True Class	Non-target	1,712,090	15,422
	Target	1,419	54,503

Table 1: Confusion matrix of $T_{d|75mm}$ on $D_{test|75mm}$

accuracy(% correct classification) :

$$\frac{1,712,090 + 54,503}{1,783,434} = 99.056\%$$

- **Results on the 200mm Dataset**

  Size of $D_{train|200mm}$: 64,127 feature vectors

  Size of $D_{test|200mm}$: 241,842 feature vectors

		Assigned Class	
		Non-target	Target
True Class	Non-target	231,495	1,702
	Target	254	8,391

Table 2: Confusion matrix of $T_{d|200mm}$ on $D_{test|200mm}$

accuracy(% correct classification) :

$$\frac{231,495 + 8,391}{241,842} = 99.19\%$$

5. Feature Elimination via Decision Tree Pruning

We conducted experiments to determine what morphological features did not contribute significantly to the performance of the decision tree classifier accuracy. For this purpose we trained a decision tree classifier and pruned it using different entropy threshold values determined from a cross validation dataset. The outline of this process follows:

1. Partition the entire data set into three distinct subsets $DS_A, DS_B,$ and DS_C by means of sampling while eliminating duplicate vectors.
2. Build a decision tree $T_{d|A}$ from the subset DS_A using the thresholding decision rule (as in Section 4)
3. During the first pass we use DS_B to compute the entropy E at each node N of $T_{d|A}$ as follows:

   $$E_N = -k_{0,N} log\left(\frac{k_{0,N}}{k_N}\right) - k_{1,N} log\left(\frac{k_{1,N}}{k_N}\right)$$

   where $k_{0,N}$ represents the number of vectors with label 0 (non-target) at node N, $k_{1,N}$ the number of vectors with label 1 (target) and $k_N = k_{0,N} + k_{1,N}$.
4. During the second pass the entropy change at *non-leaf* nodes N is determined as:

   $$\Delta E_N = E_N - \frac{k_L}{k_N} E_L - \frac{k_R}{k_N} E_R$$

   where k_L (k_R) represents the number of vectors the threshold decision rule sends to the *left (right)* and $E_L(E_R)$ represents the entropy of the left (right) child node of node N in $T_{d|A}$.
5. During the final pass every internal node N that meets the criterion

   $$\Delta E_N < \mathcal{E}_\theta$$

   is removed along with its left- and right-subtrees. Where $\mathcal{E}_\theta$ is an input parameter indicating the minimum entropy change allowed

Steps 2 through 5 were repeated for 18 different threshold levels $\mathcal{E}_\theta \in \{0.5, 1.5, 3.5, 5.5, 7.5, \ldots, 33.5\}$. The entire process is repeated switching the roles of DS_A and DS_B so as to find commonly eliminated features in the process, thus yielding 36 pruned decision trees.

We used each pruned tree as a classifier on DS_C and gathered the following characteristics:

- accuracy of correct classification
- *false-non-target* rate
- *false-target* rate

- features not used in the decision tree
- size of the pruned tree

Table 3 and *Table 4* show the number of vectors in each feature dataset.

Feature Dataset	#Non-target	#target	total
DS_A^{q75}	450,766	44,504	495,270
DS_B^{q75}	458,987	45,372	504,359
DS_C^{q75}	467,884	46,185	514,069

Table 3: Number of feature vectors for 75mm resolution

Feature Dataset	#Non-target	#target	total
DS_A^{q200}	75,432	7,090	82,522
DS_B^{q200}	76,530	6,956	83,486
DS_C^{q200}	78,261	7,209	85,470

Table 4: Number of feature vectors for 200mm resolution

5.1. Feature Elimination Experiment Results

The number of nodes versus the accuracy for each of the 36 pruned decision trees helps us choose a decision tree which has less nodes while the accuracy is still as high as possible. The following graph summarizes the results for the dataset based on 75mm resolution E3D images.

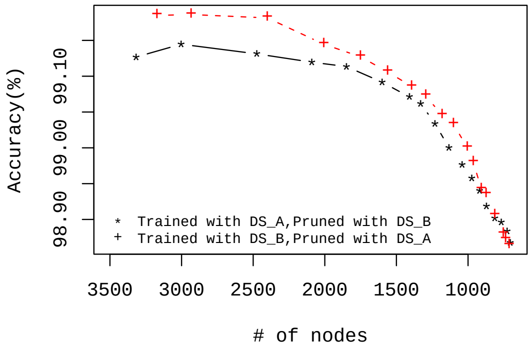

Figure 2: Number of nodes vs. classification accuracy(75mm)

As it can easily be noted from *Figure 2*, we were able to prune the decision tree while still keeping the accuracy to a reasonable level. This also speeds up the classification process on new images since we do not have to create those features which were eliminated based on the entropy change criterion.

We achieved similar results on the dataset based on 200mm resolution E3D images *Figure 3*:

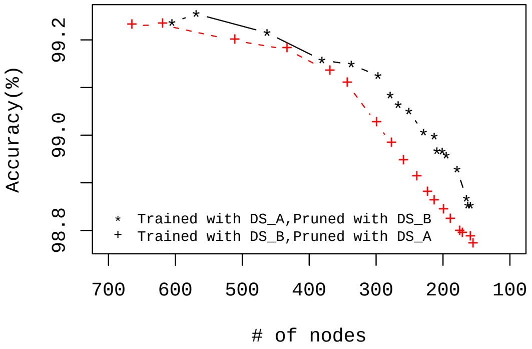

Figure 3: Number of nodes vs. classification accuracy(200mm)

6. Conclusion

We present an Automatic Target Detection system that works on 3D range images. We treat the 3D data as if the height above the ground values are gray scale intensity values and process them using 2D mathematical morphology operations to generate features characterizing the shape of the objects in the scene. Since different types of objects have different height characteristics, the features we generate clearly reflect those characteristics as well as the 2D shape characteristics.

After generating a large number of morphological features efficiently we let the decision tree classification algorithm determine the decision regions. Furthermore, we perform feature elimination by means of decision tree pruning based on the entropy change at each internal node of the decision tree. The experiment results show that we can reduce the size of the tree by at least 25% while the accuracy reduces to 98.7% at minimum.

Future work will be on classifying different types of objects based on their 3D shape characteristics using the techniques we described in this paper.

References

[1] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth International Group, Belmont,CA, 1984.

[2] J. Y. Gil and R. Kimmel. Efficient dilation, erosion, opening and closing algorithms. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 24(12):1606–1617, December 2002.

[3] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision, volume I*. Addison-Wesley, Reading, MA, 1993.

[4] R. M. Haralick, S. Sternberg, and X. Zhang. Image analysis using mathematical morphology. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 9(4):532–550, July 1987.