Improvement in Range Segmentation Parameters Tuning

Luigi Cinque
Fabrizio Corzani
Stefano Levialdi
Dipartimento di Informatica
Università di Roma “La Sapienza”
Via Salaria, 113 - 00198 Roma, ITALY
{cinque, corzani, levialdi}@dsi.uniroma1.it

Rita Cucchiara
Dipartimento di Scienze dell’Ingegneria
Università di Modena e Reggio Emilia
Via Vignolese, 905 - 41100 Modena, ITALY
rita.cucchiara@unimo.it

Gianluca Pignalberi
CASPUR
Piazzale Aldo Moro, 5 - 00185 Roma, ITALY
gianluca.pignalberi@caspur.it

Abstract

A great effort has been done during last years to improve range image segmentation results. The efficacy of the algorithms is affected by the parameters tuning.

In this work two well-known search techniques have been applied to this task: genetic algorithms and simulated annealing. These techniques are adopted in cascade: the former to obtain a rough seed point set and the latter to have a more precise refinement of suitable solutions.

We addressed our efforts towards the range segmenter proposed by the University of Bern, that seems to be the best in term of versatility, being able to segment planar and curved surfaces, and in term of speed and quality of the performed segmentations.

1. Introduction

During the last years the task of segmenting range images into object surfaces has been considered of great importance, and several techniques have been proposed to obtain the best performance: we can quote the range (surface) segmenters (RS in the following) presented in [6] and [4], and the improved version of the University of Bern segmenter [9].

All segmentation algorithms are characterized by a set of parameters to tune: this problem is particularly critical in range images where the quality of data (depending on the sensor), the noise, the surface orientation and the surface geometry contribute towards affecting segmentation. The choice of a set of parameters heavily affects the segmentation performances, and techniques to search such a set, given a set of range images and a RS, have been proposed, such as in [12] and [7].

In our previous work [2] we proposed a search tool based on genetic algorithms [5], and some preliminary results obtained applying it to the University of Bern segmenter have been presented in [3].

Is our opinion that genetic algorithm performance could be improved by other search strategies, and to empirically prove that we applied simulated annealing strategy [10] to refine the results given by the genetic search. Combining such strategies in a master-slave configuration (as showed in figure 1) we obtained better results than that allowed by the older method while reducing the total number of segmentations per image.

1.1. Range image segmentation

Several RS algorithms have been proposed to subdivide range images into surface patches [6, 11, 4, 8, 9]. They
are based on different algorithms, such as edge filling, clustering, region growing. They are characterized by a set of parameters heavily affecting their performances, mainly in terms of segmentation quality. A good set of parameters could affect also the segmentation speed, avoiding unnecessary steps during the process.

While several RS have been presented till now, we believe that their potential capability has not been explored. New techniques are proposed and compared to the older (as in [4]), but apparently only few attempts were made to assess the real capability of formerly built segmenters (such as in [6, 12]).

We focused our attempt to explore RS potential on the segmenter of the University of Bern [6, 9] (UB from now), slightly modified at the University of Modena to segment synthetic images of the MSU database [1]. We firstly built a system based on genetic algorithms [5] that was able to explore the solution space to find a suitable set of parameters for that segmenter. The promising results obtained encouraged us to improve the search strategy, so we decided to use the output of the genetic search as a starting point for a simulated annealing process [10]. In this way it should be possible to obtain a more suitable solution at the cost of a relatively small increase of computation.

The paper is structured as follows: in section 2 a brief description of the project and of the used algorithms is presented; in chapter 3 are shown some experimental results; lastly in section 4 a little discussion is proposed. A succinct bibliography is given at the end of the paper.

2. The proposed approach

The approach we used in the present work is to combine two search techniques to obtain a better result than using only one of the two strategies.

As previously stated, and shown in figure 1, we use genetic algorithms to find a rough but selected set of candidate solutions, and then, through a Simulated Annealing process, we try to refine the quality of the previously proposed results [2, 3].

2.1. First step: Genetic Algorithm

Genetic algorithm (GA from now) is a search strategy based on the paradigm of natural evolution. Its fundamentals are today quite well known, but they are still objective of study and research.

GAs have a common structure: they start from an initial population (i.e. a set of individuals –possible solutions– that are the basis to whom apply evolution); each individual has a chromosome (a data structure containing the characteristics of an individual) and its goodness is evaluated by a fitness function (the function scoring the fitness of an individual within the considered environment). New individuals are generated by the crossover (i.e. the function that performs the coupling of two individuals and the following generation of children sharing some characters of the parents).

The chromosome we adopt is a juxtaposition of the 10 parameters of UB (either integer or fixed point), and fundamentals of GA we adopted are better described in [2]; in this work we modified the running parameters according to table 1, where we see two used crossovers: ROFD and RORD (plus SPC, single point crossover [5]. They are both based on the principle that a couple could breed a different number of children (and not always two, as commonly assumed). Therefore they are able to generate a number of children one through seven children according to a fixed probability distribution and according to division schema (fixed for ROFD and random for RORD), as described in [2].

2.2. Second step: Simulated Annealing

Simulated Annealing (SA from now) is a search technique inspired by the homonymous physics experiment, presented in [10]. It is a slight modification of hill climbing technique [13]. The experiment SA wish to carry out is to melt a solid, reaching a certain temperature, and then decreasing the temperature to reach a final solid state having minimum energy. Since it could be possible to pass through a high energy state during the process, we have to foresee that probability, showed in equation 1,

\[ p = e^{-\frac{\Delta E}{k_B T}} \]  

where \( \Delta E \) is the energy difference, \( k_B \) is the Boltzmann constant and \( T \) is the temperature.

In our case \( \Delta E \) represents the fitness difference (where the fitness function of the GA is exactly the cost function of the SA, and then we will call it \( \Delta C \)), and \( k_B T \) is a quantity that we can fix by heuristics for every step of the experiment; all that quantities will compose a table of temperatures, called, for convenience, annealing table (\( T_A \)). Obviously \( T_A \) contains several entries, to which we will refer using lowercase sub-indices. Thus the used equation will be:

\[ p' = e^{-\frac{\Delta C_i}{T_A i}} \]  

where \( \Delta C_i = C_i - C_{i-1} \) (i representing a time); \( p' \) represents the probability to pass to a new state even if the current

<table>
<thead>
<tr>
<th>Table 1. Working parameters for GA step</th>
</tr>
</thead>
<tbody>
<tr>
<td># of gener.</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>Prev. work</td>
</tr>
<tr>
<td>Curr. work</td>
</tr>
</tbody>
</table>
Table 2. Working parameters for SA step

<table>
<thead>
<tr>
<th>$T_0$</th>
<th>$T_f$</th>
<th>$\alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.01</td>
<td>0.9</td>
</tr>
</tbody>
</table>

cost is worse than the previous.

The tuning of the annealing table is a task to perform very carefully, because it could be possible to waste a lot of CPU time when the process has reached an optimum, or to reach very early a local optimum without the possibility to improve the quality of the solution. In equation 3 is reported the decreasing type rate of temperature and in table 2 we find the values for equation 3, which make us able to reconstruct the annealing table.

$$T_{A_{i+1}} = T_{A_i} \cdot \alpha$$

(3)

The search is performed by SA according to the following rationale: a whole annealing cycle is completed varying the first of the ten parameters of UB. Once we get a solution we perform another annealing cycle varying the second parameter, fixing the first, and so on till the tenth.

The solution given by this process is accepted as the final solution.

3. Experimental results

The obtained results are compared to that partially presented in [2] and [3].

In figure 2 we can see the plot of the obtained scores for the 35 images used as reference set (numbered according to the alphabetical order). It is straightforward that the GA+SA strategy obtains almost always better results than the GA-based strategy and the human choice of parameters.

In table 3 we can see the obtained data relatively to the number of iterations and the fitness (only average values due to space limitation). These data are directly compared to those obtained by applying GA to the segmentation (as in our previous works) and to those, called heuristic, judged satisfactory by the implementer of the segmenter. Average execution times are not presented because we used different reference machines to evaluate experiments. Only six families of images (30 images) are presented because the seventh is composed by images belonging to two families (hump and taperoll).

From table 3 we can see that the GA+SA strategy, during the GA step, performs about 65% of the segmentations required by the “GA only” strategy. Moreover we have an average fitness about 10% better.

As a visual example of improvement we can see figure 3 where a segmented image is presented with the corresponding range image. In raster scanning order we see the range image, the segmentation obtained using the human-selected parameters, the segmentation by GA-selected parameters and the same image but segmented by GA+SA-selected parameters.

As we can see the image segmented by human-selected parameters is less precise, presenting less accurate borders and a great number of “lost” points. The result given by the GA-selection is more accurate and has more precise surface patches. The GA+SA selection gives much more precise results, with few “lost” points and very well defined surface patches, even if it fails in finding a surface (as the other selected parameters sets).

4. Discussion and conclusions

In the present work we focused again our attention to the range image segmentation process and to the parameters tuning task, looking for a strategy to improve the quality of the research and to shorten the required computational time.

Since GAs are known to be slow in convergence but are able to assure adequate results, we used them to restrict the solution space to a certain number of seed points. These points will subsequently be processed by a SA, that will look for a better solution slightly perturbing the solutions proposed by the GA. In this way we are able to shorten the number of iteration of the GA and, adequately tuning the SA parameters, have a good tradeoff between required number of iterations and found solution. The obtained results are promising, and also better than that found in our previous works ([2] and [3]).

Unfortunately at the moment we cannot compare our method to the most similar one [12] because of different databases and segmenters used. But such a test should follow.
Table 3. Average number of segmentations and fitness obtained by the compared methods

<table>
<thead>
<tr>
<th>Family</th>
<th>Avg. # of segmentations</th>
<th>Avg. fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Heuristic</td>
<td>GA only*</td>
</tr>
<tr>
<td>Adapter</td>
<td>N.A.</td>
<td>614.7</td>
</tr>
<tr>
<td>Agpart2</td>
<td>N.A.</td>
<td>1005.7</td>
</tr>
<tr>
<td>Bigwye</td>
<td>N.A.</td>
<td>859.9</td>
</tr>
<tr>
<td>Cap</td>
<td>N.A.</td>
<td>967.0</td>
</tr>
<tr>
<td>Column1</td>
<td>N.A.</td>
<td>2124.9</td>
</tr>
<tr>
<td>Curvblock</td>
<td>N.A.</td>
<td>2122.6</td>
</tr>
</tbody>
</table>

*Average value obtained using both crossovers
**Total number of segmentations; 498 is the number of segmentations of SA step

Figure 3. Example of segmentation quality. From top-left to bottom-right: original range image, human parameters, GA parameters, GA+SA parameters

References