Bulletin of the *Transilvania* University of Braşov • Vol 13(62), No. 2 - 2020
Series III: Mathematics, Informatics, Physics, 697-704
https://doi.org/10.31926/but.mif.2020.13.62.2.25

ALGORITHM FOR MERGING AND INTERPOLATING CLUSTERS IN OVERLAPPING IMAGES

Laura CIUPALĂ[1], Adrian DEACONU[2] and Delia SPRIDON[*,3],

Abstract

An image overlapping algorithm, taking into account certain properties of objects identified in the images (average intensity, movement speed, etc) is proposed. The algorithm minimizes both memory and time complexity and it can be used in various applications, especially in medical imaging analysis. The idea behind the proposed algorithm is surface merging and interpolation.

2000 *Informatics Subject Classification:* 68U10, 54H30, 68Q25.
Key words: medical imaging, image processing, polynomial complexity

1 Introduction

Image processing algorithms and computer vision are continuously studied, developed and improved due to their large set of potential applications in various domains such as medical diagnosis, photography, cinematography or industrial aspects[1, 2, 3]. One of the most important domains where image processing is crucial is medical imaging. Medical imaging plays an increasingly important role in multiple clinical scenarios, such as diagnosis or treatment planning. Thus, it is needed to avoid the subjectivism of visual human inspection when the tumours, organs or other target appearances are analysed, such as sizes, shapes, textures, evolution, and to continuously develop algorithms that are able to process immense quantities of data helping the physicians to take the best decision for the pacient. There are already developed a wide variety of techniques for medical image processing to identify and analize the targets that need to be studied such as: cells, tumours, tissues etc [4, 5]. The obvious advantages of computer-aided analysis are objectivity and an increased processing speed. Therefore, a reduction in cost, a

[1]Faculty of Mathematics and Informatics, *Transilvania* University of Braşov, Romania, e-mail: l.ciupala@unitbv.ro
[2]Faculty of Mathematics and Informatics, *Transilvania* University of Braşov, Romania, e-mail: a.deaconu@unitbv.ro
[3*] *Corresponding author*, Faculty of Mathematics and Informatics, *Transilvania* University of Braşov, Romania, e-mail: delia_spridon@yahoo.com

smaller probability for a false detection and an increased number of information about certain target properties.

In this paper, we focus on an algorithm that can be used for merging multiple images, combining properties of each target from individual frame and obtaining a final image that contains all overlapped targets. The algorithm can be used to recompose 2D image from tomografic slides in order to calculate tumoral surface or maximum tumoral diameter, for a better analysis and diagnosis based on medical imaging.

The paper is organised as follows: Section 2 describes the proposed algorithm and, also, the proposed preudocode of the algorithm is presented and detailed. In the last section the conclusions are drawn.

2 Algorithm description

There are already developed algorithms to objectively identify tumours or other targets in medical images: classic algorithms [6] or machine learning algorithms [7]. Tumour regions are automaticaly identified and extracted from images obtained by using various medical techniques such as Magnetic Resonance Imaging (MRI) or Computer Tomograph Images (CT) [8, 9]. Segmentation, edge detection algorithms [8] or Artificial Neural Network[9] are usually used for a fast and good tumour identification. Usually, medical techniques give images of slices through the inspected region. After obtaining the section in each frame the properties of the entire reconstructed target need to be studied for a better understanding of the evolution and for establishing the best treatment procedure. Thus, the maximum diameter, surface, volume and even increasing speed from one investigation to the next, etc. need to be found and studied for the entire reconstructed target. In order to do this it is necessary to be also able to rebuild the target image taking into account the properties from each image slice of the medical procedure. Therefore, for each image the clusters / targets (that can be tumoral formations, but also cells, tissues or any other formations that are needed to be investigated based on studied images) are identified using known algorithms for contour identification [8]. Our algorithm rebuilds the 2D tumoral image taking into account the image from each slice and interpolates the properties of all overlapping tumour or cluster frames.

In order to obtain all intersecting clusters the algorithm follows a few steps. First the tumoral regions are identified in each image frame using known algorithms for tumour identification and also their properties are analyzed. Secondly for each frame an array of cluster structure that contains information about tumour properties in the corresponding frame is built as well as a negative index that will point to the corresponding tumoral regions in the final image. Also, a corresponding 2D array is built and each of its elements contains the index of the cluster structure in the first array. Based on the identified clusters an array of clusters for each new image / frame and a matrix of indexes are built. Therefore, for each pixel (i,j) of the image, an element (i,j) corresponds in the matrix with the address of the corresponding cluster position in clusters' array. A new matrix is composed by overlapping frame by frame the studied images and a final matrix is obtained having on each element the address of an array's element, that also points to the address of all intersecting clusters. In the end, an array of cluster structures is obtained that has the length equal to the total number of tumoral regions in all the frames, an array of sets of addresses (each address points to the first array) and a 2D array where each element is an index of the first array.

In Figure 1 an example of two overlapping images (Figure 1a and Figure 1b), the multiple cases of possible cluster intersection (Figure 1c) and also the final image (Figure 1d) are presented. The three possible intersection cases are as follows:

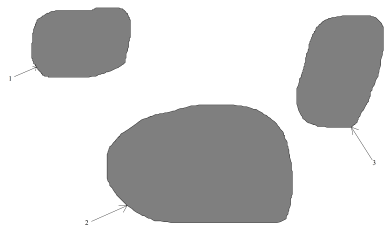
- **Case 1.** The new cluster does not intersect any other already added clusters (from previous frames) - cluster "a" in Figure 1c. In this case, the new cluster info is added to the cluster array and all matrix elements in the resulted image, that corresponds to the cluster's pixels point to the new added cluster.

- **Case 2.** The new cluster intersects another cluster and it was not added to the new array - cluster "b" in Figure 1c. In this situation, the new cluster info has to be interpolated with the already added cluster's info, and all the matrix elements corresponding to the new cluster in the resulted image are pointing to the interpolated cluster image.

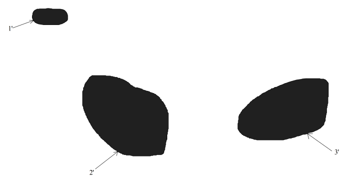
- **Case 3.** The new cluster intersects another cluster and it was already added to the new array - cluster "c" in Figure 1c. The cluster info that has already been added has to be interpolated with the new intersected cluster info, thus, all the elements of the matrix that have already been set to point the current cluster have to be set to point to the intersected cluster info. This case is more complex because the previous cluster's pixels need to be found so that they point to the new info of the composed cluster. In order to avoid such search that is very much time consuming, only the value from the pointed address has to be modified, so that both the new cluster and the old cluster have the same info.

As it was mentioned before, in order to reduce the complexity of the algorithm 2 arrays are used to store the links between different clusters of the added image and the main cluster from the final image. The elements of the first array are the clusters from all images that are interpolated, and each element of the first array contains the cluster property and an index of an element in the second array. The elements of the second vector are added when a new cluster is found in the added image that does not exist in the previously added images and each element of the second array is a set of addresses of the cluster structure that belongs to the same cluster in the final image.

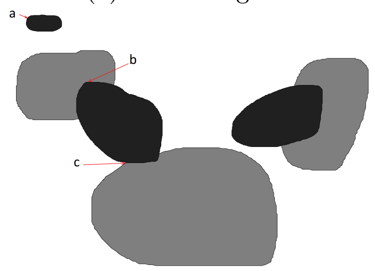
Thus, after identifying clusters and their properties for each new image this information is added to the first vector. Also, a corresponding 2D grid is saved.



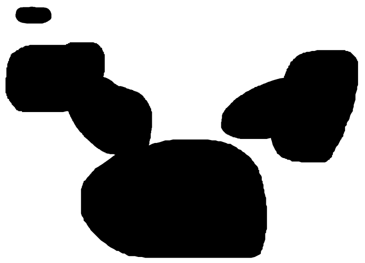
(a) First image



(b) Second image



(c) The clusters' intersection cases

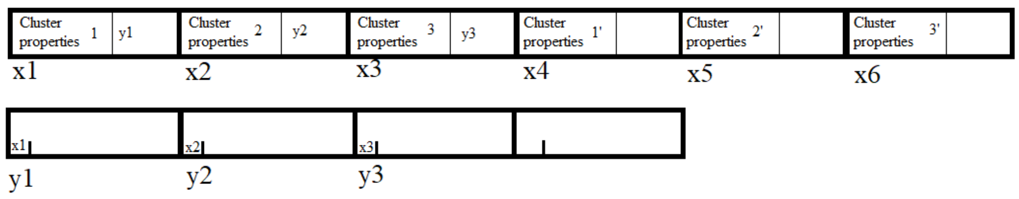


(d) Final image

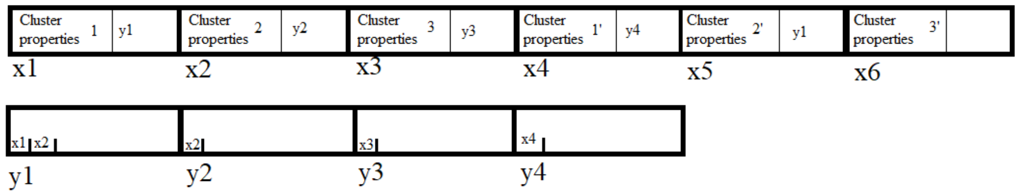
Figure 1: Example of two overlapping images

This grid contains the position for each pixel of the cluster to which it belongs. Each new 2D grid is analyzed pixel by pixel. When a new cluster in the new image is found, its corresponding cluster structure is added and if an intersection with an older cluster is not identified a new element in the second array with the address of the new cluster element is also added. In cluster structure the address of this element in the second array is stored. If a new intersection between the current cluster and an older one is found a few steps need to be performed depending on the situation (see Figure 2):

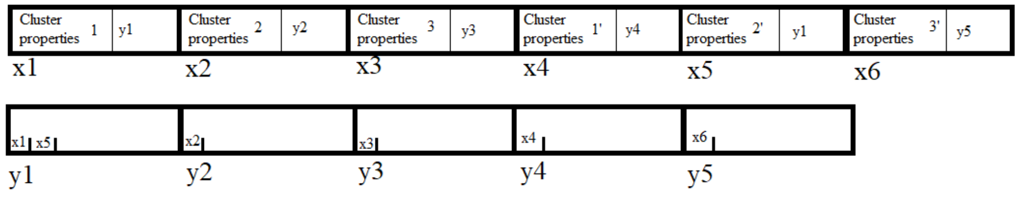
- If the cluster address was not yet added to the second array, the current cluster address is added to the vector of the previous corresponding element in the second array and the address from the cluster's structure will point to the respective element (Figure 2b, Figure 2c).
- If the cluster address was already added to the second array (Figure 2d, Figure 2e), then the following actions are performed:
  - each cluster structure pointed by the second array's element (including the current cluster structure) is checked and their index is replaced with the older intersected cluster corresponding index;
  - all addresses from the second array's element are moved to the one from the previous intersection;
  - the set of addresses from the moved element is cleared.



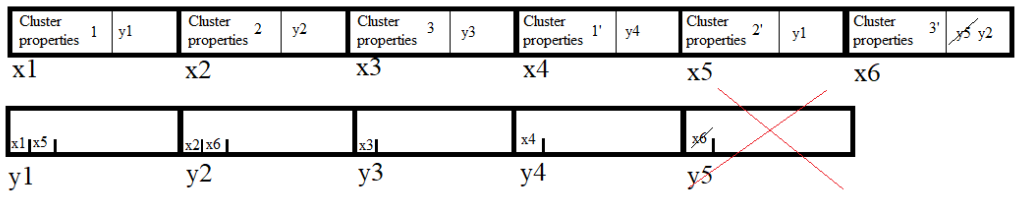
(a) The two arrays after adding the first image info



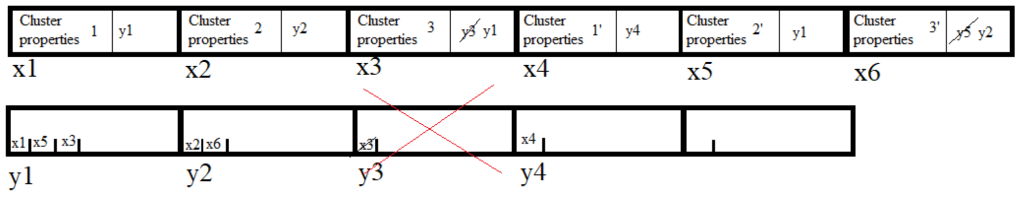
(b) The two arrays after finding the first cluster structure in the second image and the second cluster structure in the second image, intersecting with the first cluster structure from the first image



(c) The two arrays after finding the third cluster structure in the second image



(d) The two arrays after finding the second cluster structure in the second image, intersecting with the first cluster structure from the first image



(e) The two arrays after finding the first cluster structure in the second image

Figure 2: The evolution of the two arrays during the second analysis of the image

Pseudocode of the algorithm

```
struct ClusterStructure {
   Property clusterProperty;
   int noOfPixels;
   int ix;
}
AddNewFrame(newIm, noPrCl, &allClV, &clLk, &fIm)
{
for (i=0; i < n; i++)
   for (j=0; j < m; j++)
   {
      int posInfo = newIm[i][j] + noPrevCl;
      if (newIm[i][j] != 0)
      {
         if (fIm[i][j] == 0 && allClV[posInfo].ix < 0)
         {
            set<ClusterStructure*> newClusterAddress;
            newClusterAddress.push_back(& allClV[nIm[i][j]]);
            clLk.push_back(newClusterAddress);
            allClV[nIm[i][j]+ noPrCl].ix = clLk.size() - 1;
            fIm[i][j] = nIm[i][j];
         }
         else if (fIm[i][j] != 0)
         {
            if (allClV[posInfo].index < 0)
            {
               clLk[fIm[i][j].index].insert(&allClV[posInfo]);
               allClV[posInfo].index = fIm[i][j];
            }
            else
            {
               int pos = allClV[posInfo].ix;
               int noIntersectedClusters = clLk[pos].size();
               for (k=0; k < noIntersectedClusters ; k++)
               {
                  clLk[pos][k]->ix = allClV[fIm[i][j].ix];
                  clLk[fIm[i][j].ix].push_back(clLk[pos][k]);
               }
                clLk[pos].clear();
            }
         }
      }
   }
```

In the above algorithm the used variables are as follows:

- `nIm` - the matrix corresponding to the current image frame that needs to be added
- `noPrCl` - total number of clusters of the previous added frame images
- `allClV` - the array of all clusters from each image
- `clLk` - the linker array (the array that contains on each position a set of addresses to all clusters that intersect from the `allClV` array)
- `fIm` - the matrix corresponding to the recomposed image (final image)
- `n, m` - the image dimensions (height and width)

The properties of each cluster can be averaged by iterating the second array, and, for each element, a mean or weighted average of certain property can be calculated.

3 Conclusions

In this paper, an efficient algorithm is proposed that can be successfully used to calculate the average of certain properties of various targets from a set of images obtained from medical procedures such as MRI or CT. The medical procedures return slices through tumoral regions or tissues and they are combined to obtain a 2D final image putting together all the information.

References

[1] Pisano, E.D., Cole, E.B., Hemminger, B. M., Yaffe, M.J., Aylward, S.R., Maidment, A.D.A., Johnston, R.E., Williams, M.B., Niklason, L.T., Conant, E.F., Fajardo, L.L., Kopans, D.B., Brown, M.E. and Pizer, S.M., *Image processing algorithms for digital mammography: A pictorial essay*, Radio Graphics **20** (2000), 1479-1491

[2] Bi, S., Chen, H., Ke Y., Rao, S. and Liu, J., *Processing algorithms for quantum remote sensing image data*, Proc. of SPIE 2019, Infrared Remote Sensing and Instrumentation XXVII **11128**, San Diego, California, USA, August 2-14, 2019 (Eds. Marija Strojnik and Gabriele E. Arnold), Curran Associates, Inc., 2019

[3] Chen, Y., *Image processing algorithms of Hartmann aberration automatic measurement system based on tensor product network*, Int J Wireless Inf Networks **26** (2019), 158–164

[4] Antonie, M.L, Zaïane, O.L., and Coman A., *Application of Data Mining techniques for medical image classification*, Proceedings Second International Workshop on Multimedia Data Mining, August 26th San Francisco, California, USA, 2001

[5] Karabağ, C., Jones, M.L., Peddie, C.J., Weston, A.E., Collinson, L.M. and Reyes-Aldasoro C.C., *Semantic segmentation of HeLa cells: An objective comparison between one traditional algorithm and four deep-learning architectures*, PLoS ONE **15(10)** (2020),

[6] Gulo, C.A.S.J., Sementille, A. and Tavares, J.M.R.S., *Techniques of medical image processing and analysis accelerated by high-performance computing: A systematic literature review*, Journal of Real-Time Image Processing **16** (2019), no. 6, 1891-1908

[7] Linder, N., Taylor, J.C., Colling, R., Pell, R., Alveyn, E., Joseph, J., Protheroe, A., Lundin, M., Lundin J. and Verrill, C., *Deep learning for detecting tumour-infiltrating lymphocytes in testicular germ cell tumours*, Journal of Clinical Pathology **32** (2019), no. 2,157-164

[8] William Thomas, H.M. and Kumar, S.C.P., *A review of segmentation and edge detection methods for real time image processing used to detect brain tumour*, IEEE International Conference on Computational Intelligence and Computing Research (ICCIC) 2015, Madurai, 2015, 1-4

[9] Chithambaram, T. and Perumal, K., *Edge detection algorithms using brain tumor detection and segmentation using artificial neural network techniques*, International Research Journal of Advanced Engineering and Science **1** (2016), no. 3, 135-140