Efficient Multi-scale Patch-Based Segmentation

Abinash Pant[1,2(✉)], David Rivest-Hénault[1], and Pierrick Bourgeat[1]

[1] CSIRO Digital Productivity Flagship, The Australian e-Health Research Centre,
Herston, QLD, Australia
{David.Rivest-Henault,Pierrick.Bourgeat}@csiro.au
[2] University of Burgundy, Le Creusot, France
Abinash_Pant@etu.u-bourgogne.fr

Abstract. The objective of this paper is to devise an efficient and accurate patch-based method for image segmentation. The method presented in this paper builds on the work of Wu et al. [14] with the introduction of a compact multi-scale feature representation and heuristics to speed up the process. A smaller patch representation along with hierarchical pruning allowed the inclusion of more prior knowledge, resulting in a more accurate segmentation. We also propose an intuitive way of optimizing the search strategy to find similar voxel, making the method computationally efficient. An additional approach at improving the speed was explored with the integration of our method with Optimised PatchMatch [11]. The proposed method was validated using the 100 hippocampus images with ground truth segmentation from ADNI-1 (mean DSC = 0.892) and the MICCAI SATA segmentation challenge dataset (mean DSC = 0.8587).

1 Introduction

Proliferation of atlas images along with the growing availability of computational resources has made multi-atlas based segmentation techniques popular. These techniques primarily revolve around using multiple atlas images to introduce expert knowledge into the segmentation process. These atlases are registered to the target images space and the information is propagated to determine the label of the target image. Heckemann et al. [1] pioneered such techniques using 30 normal brain MR images providing accurate segmentation. Aljabar et al. [2] proposed improvements by selecting only a subset of the complete atlas set that are more anatomically similar to the target image. In contrast to global selection of atlases, various patch-based techniques [6,7] that look at patch wise similarity have gained popularity. A local weight is computed for each of the local patch extracted from the atlas. These computed weights are based on intensity similarity with the target patch and are then used in the label fusion process to produce the final segmentation. Sparse representation, with its success in face recognition [4], have been extensively used [8,9,14] to estimate these weights.

However, calculating these weights can be very costly especially when larger 3D patches are required to capture the appropriate level of information. Pre-selections strategies for discarding atlases or patches [5,6] have been employed

to alleviate this problem. Even though these strategies ease the process of calculating weights, they discard information, which could lead to sub-optimal results [11]. The use of large patches introduces another problem in segmenting small structures, as pointed out by [14]. Patch similarity of large anatomical structures could misguide the labeling of small structures around it.

Another common technique used in patch-based methods is to define a search window, around the target voxel, where similar patches are searched. They are used to increase the accuracy of the method as similar patches would lie in the vicinity of the target voxel in a well registered atlas. This also provides a computational boost as the search is limited to a small window rather than the whole image. However, these windows are constant and do not evolve depending on the knowledge provided by the atlases or the anatomy of the structure under consideration. This results in identical effort being exerted in calculating weights for each patch. But in reality the effort required is not uniform and can be approximated using prior information available. For example effort required to label a patch near the centre of a solid sphere should be much less, as a majority of atlas labels would belong to sphere class, than a patch which is closer to its surface. Similarly a large search window might adversely affect the segmentation result of a small well registered structure, by bringing in more outliers.

In this paper, we use a compact multi-scale feature to avoid the use of pre-selection strategies while still being discriminative. We also propose a search criterion that evolves according to the confidence level of a region provided by the atlas, to lower the execution time. We have also integrated our method with Optimised PatchMatch as an additional approach to make our method computationally efficient. Our method is validated against the ADNI and MICCAI SATA data set. A comprehensive validation is performed to show the efficacy of the proposed enhancements.

2   Method

The main objective of any multi-atlas segmentation approach is to estimate a label map L_t for a given target image I_t. This is achieved by using a set of N atlas of images $I = \{I_s | s = 1, 2 \cdots I_N\}$ and their corresponding label map $L = \{L_s | s = 1, 2 \cdots N\}$ which is registered to the target image. For each voxel x ($x \in I_t$) a set of Q voxels are selected from the atlas around a search neighbourhood $n(x)$. A feature vector v_b ($b = 1, 2 \cdots Q$) is created for each of these voxels and assembled column wise into a matrix V. A set of weights w_b is estimated to reconstruct the target voxel's feature vector A using V. The label for the target voxel $L_t(x)$, out of M possible labels, can be calculated by $\arg\max_{m=1\cdots M} \sum_{b=1}^{Q} (w_b.\delta(l_b, L_m))$ where l_b is the label corresponding to each feature vector v_b. This gives a general overview of the method used for segmentation. Details of each component after the registration step has been explained below.

2.1   Multi-scale Feature

Patch-based techniques traditionally use only a single scale patch, which puts a hard limit on the available information. These methods [6,7,9] generally use a larger patch to compensate for the lack of high level information which leads to an increase in computation time. Multi-scale approaches have been proposed [14] to increase the prior information, but, somewhat surprisingly, they dedicate a majority of the voxels to represent the coarser scale making them bulky. We argue that there is no compelling benefit in assigning a large number of voxels to represent these coarser scale. It is especially true when these scale are not crucial in determining the final label of the target voxel and a downsampled version could prove to be equally effective. To overcome this problem we use a simple multi-scale patch consisting of δ patches (one per scale) of size $\epsilon \times \epsilon \times \epsilon$ at different scales. It can also be naturally extended to include variable patch size for each scale. For a typical feature vector ($\delta = \epsilon = 3$) the size of the vector would be 81, which is an order of magnitude smaller than the vectors used in related work [14].

The effectiveness of the proposed feature vector is illustrated by a simple example shown in Fig. 1, which shows the response of different feature vectors of the emphasized voxel over the whole image. We can observe that multi-scale patch response (c) and (d) is more precise and localised than that of single scale patch response (a) and (b). It is also evident that the response of our patch (d) is much more targeted than that of (c). This may be attributed to the fact that the patch used in (c) uses a Gaussian filter at the original scale without sub-sampling thus would require a much larger patch to encode the same level of information as compared to ours (d).
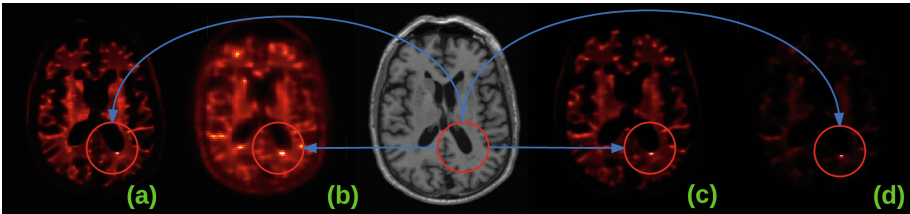
Fig. 1. Response of the feature using sum of squared differences. (a) Small scale patch (b) Large scale patch (c) Multi-scale patch of Wu et al. [14] (d) Our multi-scale patch

Patch-based techniques by design are computationally very expensive therefore most of the proposed methods use pre-selection techniques to limit the atlas/patches to make it computationally efficient. As pointed out by [11], this reduces the problem scope, which might lead to less desirable results. The size of the feature vector also contributes to the computational burden, thus using a light weight feature vector allows to include rich information while staying computationally efficient.

2.2   Sparse Label Fusion

Sparse representation is the method of representing an input data using a linear combination of a small part of an over-complete dictionary leading to a compact representation. Such representation has been proven very successful in classification applications such as face recognition [4]. The use of these techniques, for multi-atlas segmentation, was first proposed by [8,9]. Here the target feature vector v_t can be represented as a combination of the selected vectors v_b weighted by factor w_b such that $v_t = \sum_{b=1}^{Q}(w_b.v_b)$. The whole system can be represented as $A = V \times W$ where W is the matrix formed by concatenating the weights. This being an under-determined system does not have a unique solution. Therefore, a sparse constrain $min\|W\|_1$ is added, which converts the problem into a ℓ_1 minimization problem where a majority of the weights w_b are zero. The weights can be obtained using Eq. 1. The weights obtained are used to estimate the target label from atlas labels providing the final segmentation. This optimization problem has been solved using the SLEP package [3].

$$\hat{W} = \arg\min \frac{1}{2}\|A - V \times W\|_2 + \lambda\|W\|_1. \tag{1}$$

2.3   Hierarchical Pruning

During the process of finding the sparse coefficients to determine the label of the target, a hierarchical pruning step proposed by Wu et al. [14] has been used. This process has been found to improve accuracy and reduce computational burden. The basic idea of hierarchical pruning is to break the complex task of obtaining the sparse coefficients into several stages. Initially, information from all the scales is used to determine a set of candidate patches. Information from coarser scale and weak patches is removed iteratively until only patches from the finest scale remains. This ensures that final set of patches are similar to target patch at all scales and that no scale, particularly the coarse ones, dominates. This technique, with reduced iterations at earlier stage, can also act as a holistic and robust pre-selection criteria which is in tune with the overall optimization strategy.

3   Speed Improvements

In addition to computing the labels of each voxel in parallel and caching the computed feature vector v we will discuss two strategies used to increase the speed of the overall process.
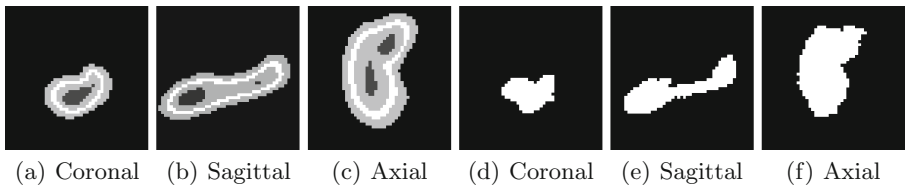
3.1   Search Optimization

Patch-based methods determine the label for a particular voxel by a weighted voting scheme where each selected voxel casts a vote depending on its label. The task becomes relatively easy when most of the voxels belong to the same class

and have the same label. In this scenario there is no need to find the exact voting power or weight of every voxel as the label of the target would be determined by the majority, regardless of the weights. Conversely the process becomes quite challenging when there is roughly an equal distribution of voxels from different classes making the weight calculation step crucial.

This information has been utilized to adaptively change the behaviour of the algorithm so that most of the computational time is spent in regions of high uncertainty. The throttling of speed is obtained by defining a search window for each level θ ($\theta \in \{0, 1, 2 \ldots\}$). The search window at each level is set to $\omega \times \omega \times \omega$ where $\omega = 2.\theta + 1$. A threshold g_θ is defined for every level θ such that each level would only handle regions which satisfies this condition $g_\theta \geq \beta > g_{\theta-1}$ where β is the percentage of majority voxels with the same label. β is calculated dynamically for each voxel, in a computationally inexpensive step, by looking at the all the atlas labels corresponding to each voxel's location. The complete search strategy is defined as $\{g_0, g_1, \ldots\}$ such that level 0 would handle regions where at-least $g_0\%$ of voxels have the same label. The remaining region would be handled by level 1 and so on. For example a search strategy of $\{100\}$ would process all patches that have the same label at level 0 and the rest of the patches are handled by level 1; this corresponds to using a window of $3 \times 3 \times 3$ in traditional approaches. The number of iterations at each of the θ levels is also set to compensate for the change in the total amount of patches being selected while increasing or decreasing the window size.

In Fig. 2(a), (b) and (c) a search optimization strategy has been created using the atlas labels. Figure 2(d), (e) and (f) show their corresponding manual delineation. The dark gray region, which denotes an easy region, would require a small search window with fewer number of iterations as the majority of voxels belong to the object. The opposite is the case for the white region, where there is an equal distribution of classes. A larger patch with a greater number of iteration is required. This strategy works very well providing faster execution time without any significant decrease in accuracy. This method is generic and can be used with any patch-based method. It can also be extended to include the anatomical information linked with the label to create a strategy tuned for a particular anatomy.

(a) Coronal   (b) Sagittal   (c) Axial   (d) Coronal   (e) Sagittal   (f) Axial

Fig. 2. Search optimization strategy ($\{95,65\}$) obtained from the atlas images (see text above). (a)–(c) shows the strategy and (d)–(f) their corresponding manual delineation

3.2   Integration with Optimised PatchMatch

Ta et al. [11] proposed Optimised PatchMatch (OPAL) that provides very accurate segmentation of MR images in a comparatively short time. The algorithm works by first generating k-Approximate Nearest Neighbour (ANN) from a given atlas and then uses a label fusion technique to obtain the desired segmentation. The algorithm runs with a constant complexity and doesn't discard any *a priori* information, the details of this algorithm have been omitted for brevity. The integration with OPAL was done primarily as an experiment to reduce the overall computational complexity by introducing a pre-selection stage in our pipeline.

To integrate OPAL within our proposed pipeline, the number k was increased while reducing the number of iteration in order to obtain a wide variety of patches which could later be fed to our method. This was done in order to reduce the probability of the same voxel being selected multiple times while still having a large collection of relevant voxels to choose from for our segmentation pipeline. This resulted in drastic reduction in overall time with comparable accuracy.

4   Results

4.1   Hippocampus Segmentation

The imaging data used for this experiment used the Harmonized Protocol[1] for Hippocampal volumetry. We used 100 T1-weighted MR images acquired following the ADNI-1 imaging protocol of 37 AD, 34 MCI and 29 Normal subjects along with their manual delineation.

For all the experiments, the atlases have been affinely registered using the robust block matching approach MIRORR [15]. To provide a fair comparison, all the other factors have been kept constant. The value for λ, used for label fusion, has been set to 0.2 for all the experiments. The values of δ and ϵ has been set to 3 giving us a feature vector that consists of $3 \times 3 \times 3$ patches at 3 different scales (1, 1/2 and 1/4 of the original image size). A 12 core Intel Xeon(R) system clocked at 3.20 GHz with 16 GB RAM was used for all the experiments.

A leave-one-out cross validation has been used to validate the proposed method. The proposed method with $5 \times 5 \times 5$ window using pruning and multi-scale feature yielded the highest Dice score of 0.892 and is used for comparison with other methods. To show the effectiveness of the proposed improvements, degraded versions of our method has been compared with this variant as shown in Table 1. We can see that multi-scale feature increased the accuracy by 0.022. Pruning provided an improvement of 0.006 in the Dice score with a slight reduction in the overall time. An improvement of 0.007 is observed when increasing the search window from $3 \times 3 \times 3$ to $5 \times 5 \times 5$.

The proposed search optimization strategy, with a configuration of {95,65}, gave a comparable result with the best variant leading to an approximate 60 % reduction in the overall time. The integration with Optimised PatchMatch, with

---

[1] http://www.hippocampal-protocol.net.

Table 1. Dice ratios and time taken for 6 variations of our method

Method	Time (s)	DSC
$5 \times 5 \times 5$ window with pruning and multi scale	**428**	**0.892**
No multi-scale	167	0.870
No pruning	689	0.887
$3 \times 3 \times 3$ window	91	0.885
Optimised search {95,65}	182	0.890
Proposed + Optimised PatchMatch ($k = 200$)	51	0.885

(a) 0.923   (b) 0.896   (c) 0.821   (d) 0.927   (e) 0.896   (f) 0.787
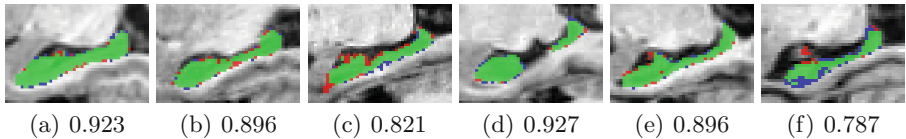
Fig. 3. Segmentation of left(a–c) and right(d–f) hippocampus of subjects with best, median and worst Dice scores (red and blue denote over and under segmentation) (Color figure online).

k = 200, provided a drastic reduction in overall time with just a slight reduction in the Dice score. A visual summary of the segmentation result provided by our method is shown in Fig. 3.

We compared our mean dice coefficients with other published results on ADNI-1 dataset. Roche et al. (BMAS) [12] and Gray et al. (LEAP) [13] reported an average dice score of 86.6 ± 1.70 and 87.6 ± 2.07 respectively on the ADNI-1 dataset. Our method yielded an average dice score of 89.2 ± 2.22 on the same dataset. It can be seen that the accuracy of our proposed method is comparable to the accuracy reported by the above mentioned methods.

4.2   MICCAI SATA Challenge

Our algorithm was validated using MICCAI SATA challenge [10]. The standardized registered diencephalon dataset containing 35 training samples and 12 testing samples were used for the validation. The images were acquired using MPRAGE (Magnetization Prepared Rapid Acquisition Gradient Echo) having a resolution of $1 \times 1 \times 1 \, mm^3$. All the images were already registered to remove any variability introduced due to the registration algorithm.

An automated system for submission provided by the challenge organisers was used to validate the result. Our method, "*HALF_R_1_4*", gave an average(median) Dice score of 0.8587(0.8696), which is just 0.0084 lower than the current leading method "*UNC MCseq*". It should be noted that this challenge does not take into account the computational complexity to evaluate the results.

5   Conclusion

The use of a smaller patch representation made it possible to include more prior knowledge which improved the overall accuracy of the method. This was validated by comparing it with a degraded version version of our method. The proposed dynamic search optimization technique was used while searching for similar patches. This approach proved to be computationally efficient and also provided comparable accuracy. A further decrease in the overall computational time was achieved by integrating our method with OPAL. Encouraging results has been obtained on the ADNI and MICCAI SATA datasets when compared with the state of the art.

References

1. Heckemann, R.A., et al.: Automatic anatomical brain MRI segmentation combining label propagation and decision fusion. NeuroImage **33**(1), 115–126 (2006)
2. Aljabar, P., et al.: Multi-atlas based segmentation of brain images: atlas selection and its effect on accuracy. Neuroimage **46**(3), 726–738 (2009)
3. Liu, J., Ji, S., Ye, J.: SLEP: sparse learning with efficient projections. Ariz. State Univ. **6**, 491 (2009)
4. Wright, J., et al.: Robust face recognition via sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. **31**(2), 210–227 (2009)
5. Coupé, P., Manjón, J.V., Fonov, V., Pruessner, J., Robles, M., Collins, D.L.: Non-local patch-based label fusion for hippocampus segmentation. In: Jiang, T., Navab, N., Pluim, J.P.W., Viergever, M.A. (eds.) MICCAI 2010, Part III. LNCS, vol. 6363, pp. 129–136. Springer, Heidelberg (2010)
6. Rousseau, F., Habas, P.A., Studholme, C.: A supervised patch-based approach for human brain labeling. IEEE Trans. Med. Imaging **30**(10), 1852–1862 (2011)
7. Coupé, P., et al.: Patch-based segmentation using expert priors: application to hippocampus and ventricle segmentation. NeuroImage **54**(2), 940–954 (2011)
8. Zhang, D., Guo, Q., Wu, G., Shen, D.: Sparse patch-based label fusion for multi-atlas segmentation. In: Yap, P.-T., Liu, T., Shen, D., Westin, C.-F., Shen, L. (eds.) MBIA 2012. LNCS, vol. 7509, pp. 94–102. Springer, Heidelberg (2012)
9. Tong, T., et al.: Segmentation of brain MR images via sparse patch representation. In: MICCAI Workshop on Sparsity Techniques in Medical Imaging (STMI) (2012)
10. Asman, A., et al.: Miccai 2013 segmentation algorithms, theory and applications (SATA) challenge results summary. In: MICCAI Challenge Workshop on Segmentation: Algorithms, Theory and Applications (SATA) (2013)
11. Ta, V.-T., Giraud, R., Collins, D.L., Coupé, P.: Optimized PatchMatch for near real time and accurate label fusion. In: Golland, P., Hata, N., Barillot, C., Hornegger, J., Howe, R. (eds.) MICCAI 2014, Part III. LNCS, vol. 8675, pp. 105–112. Springer, Heidelberg (2014)
12. Roche, F., et al.: Accuracy of BMAS hippocampus segmentation using the harmonized hippocampal protocol. Alzheimer's Dementia J. Alzheimer's Assoc. **10**(4), P705–P706 (2014)
13. Gray, K.R., et al.: Integration of EADC-ADNI harmonised hippocampus labels into the LEAP automated segmentation technique. Alzheimer's Dementia J. Alzheimer's Assoc. **10**(4), P119–P120 (2014)

14. Wu, G., et al.: Hierarchical multi-atlas label fusion with multi-scale feature representation and label-specific patch partition. NeuroImage **106**, 34–46 (2015)
15. Rivest-Hénault, D., et al.: Robust inverse-consistent affine CT-MR registration in MRI-assisted and MRI-alone prostate radiation therapy. Med. Image Anal. **23**(1), 56–69 (2015)