Optimization of Facade Segmentation Based on Layout Priors

Radwa Fathalla[1] and George Vogiatzis[2]

[1] College of Computing and Information Technology,
Arab Academy for Science and Technology, Alexandria, Egypt
[2] School of Engineering and Applied Science, Aston University, Birmingham, UK

Abstract. We propose an algorithm that provides a pixel-wise classification of building facades. Building facades provide a rich environment for testing semantic segmentation techniques. They come in a variety of styles affecting appearance and layout. On the other hand, they exhibit a degree of stability in the arrangement of structures across different instances. Furthermore, a single image is often composed of a repetitive architectural pattern. We integrate appearance, layout and repetition cues in a single energy function, that is optimized through the TRW-S algorithm to provide a classification of superpixels. The appearance energy is based on scores of a Random Forrest classifier. The feature space is composed of higher-level vectors encoding distance to structure clusters. Layout priors are obtained from locations and structural adjacencies in training data. In addition, priors result from translational symmetry cues acquired from the scene itself through clustering via the α-expansion graphcut algorithm. We are on par with state-of-the-art. We are able to fine tune classifications at the superpixel level, while most methods model all architectural features with bounding rectangles.

1 Introduction

Generating models of buildings has innumerable applications , such as heritage conservation, disaster management and urban planning. One particular field of interest has been analysis of building facades. Facades capture the architectural essence of the buildings. They are a dense representation of their characteristics in terms of layout and materials used, which translate into surface properties.

Facade parsing is often regarded as a classical case of semantic segmentation. As most scene interpretation approaches, the problem was originally tackled with appearance-based segmentation algorithms, in which weak priors of smoothness assumption are applied. Research was then directed to the incorporation of mid-level and high level cues of translational symmetry and sub-part classifications, based on training data. The challenges for achieving high accuracies rise from imaging artifacts: blur and noise, non-uniform lighting conditions, reflections and, shadows. Also, they include, the existence of irregular lattices of structures, occlusions, intra- and inter- geometric style variations.This lead to investigating the pattern of arrangement of facade elements rather than their individual visual

attributes. We present an algorithm that exploits higher level reasoning about scene entities, suggested by the appearance characteristics. We combine both aspects in a single energy function, to provide optimized solution at the lowest level of image primitives. In contrast, state-of-the-art methods [10] and [14] apply their optimization steps on formed Bounding Boxes (BB), whose assignments are either rejected or accepted as a whole. As such, their algorithms incorporates layout principles only in the recognition step of pre-segmented regions, resulting from appearance cues phase. Whereas, we carry out segmentation and recognition simultaneously, while exploiting the layout priors to correct preliminary segmentations. We provide an algorithm that minimizes the use of thresholds, prior assumptions except for fronto-parallelism and works in an approximate inference framework. More importantly, it does not require manual specification of architectural rules as in the 3-layered approach [10].

1.1 Related Work

Research is directed towards implementing architectural guidelines in automated flexible form. These guidelines are concerned with alignment, symmetry, similarity, co-occurrence and components layout. In [10], Martinović et al. make use of these architectural principles in their final classification decision. They refine the output of a preceding segmentation step by applying this set of restricting principles in an ad-hoc procedure. Each principle is applied in isolation and in most part, as a matter of fulfilling a certain criterion is exceeding a manually specified threshold. The classification into structures is achieved by an Recurrent Neural Networks RNN [12] fed with an oversegmentation of the image and a Dollar's Integral Channel [5] specialized window and door detector.

[6] is the only reported work that allows a per-pixel final classification. Every pixel is represented by a vector of image features (such as: location, RGB values, and HOG features), in addition to contextual ones (such as: neighbourhood statistics, and bounding box features) obtained from the preliminary predictions based on image features. The drawback is, each feature vector is supplied independently to an ensemble of classifiers. It lacks the concurrency in classification of pixels of the arrangement and hence, it lacks the global optimality in the proper sense. Perhaps the most related work to ours is [14].In [14],they build a factor graph of higher order cliques on the images, based on structural aspects more sophisticated than spatial proximity. However, their nodes are Bounding Boxes (BBs) of preliminary segmented regions with the pixel assignment done as a region-to-pixel mapping of the chosen label without the capability of fine tuning the results. Also, based on their reported inadequacy in localizing segment borders, the hardwired specification of thresholds on aspects like alignment, size similarity and regular spacing, will fail with inaccuracies in the segments and subsequent BBs formation. The way they handle size variations and the subsequent reliability of relative location priors is unclear, given that they use vertical and horizontal distances in their absolute form. In addition, their algorithm does not incorporate appearance in determining edges between the BBs, as they rely on purely geometrical properties.

2 Facade Segmentation Optimization

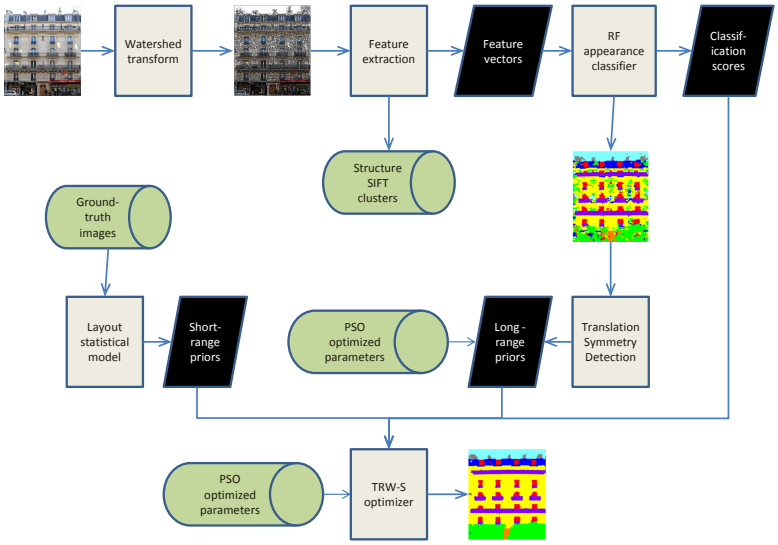


Fig. 1. Diagram showing proposed system modules and their interactions.

Our proposed algorithm (Fig. 1) receives as input a set of image pixels in the $2d$ space. It is required to provide an interpretation of these data points by assigning them to a predefined set of labels $\mathcal{L} = \{L_i\}_{m=1}^{M}$, such that $\mathcal{L}$ holds indices to M architectural structures. To keep the problem tractable and enhance computational efficiency, we work with superpixels. Thus, the data points for our algorithm is the set $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{n}$ of n superpixels. The image is subjected to a watershed transform [15]. The transform aggregates pixels to a region until reaching a peak in the $2d$ space of the gradient image. The result is a severe oversegmentation of the images with color coherent regions, called *basins*. The superpixels are the minima pixels corresponding to the lowest gradient value in each region.

We pose our problem as an optimization problem under both appearance and layout constraints, emerging from architecture characteristic patterns. To this end, we define an energy function and minimize it using the sequential tree-reweighted message passing (TRW-S) [7]. We chose this minimization technique due to its ability to handle arbitrary forms of cost function and scalability, while providing state-of-the-art results in some applications. We aim to ensure that the labeling of a pixel is influenced not only by the labeling of its neighbours,

but also by that of pixels in other possibly distant regions based on extracted architectural patterns.

A distinctive aspect of our algorithm is imparting structural knowledge on image primitives. The TRW-S operates on the original set of superpixels. The total energy function Ξ of the TRW-S is as follows:

$$\Xi = \Xi_1 (L) + \Xi_2 (L) \quad , \tag{1}$$

where

$$\Xi_1 (L) = \sum_{x_i} D (L_i | x_i) \quad , \tag{2}$$

is the datacost received from the appearance module. $D (L_i|x_i) = -\log (P (L_i|x_i))$. $P (L_i|x_i)$, are the classification posteriors resulting from a Random Forest (RF) classifier. And, the layout prior

$$\Xi_2 (L) = \beta_1 \sum_{x_i} \sum_{x_j \in \Psi_1} Q_1 (L_i, L_j | x_i, x_j) + \beta_2 \sum_{x_i} \sum_{x_j \in \Psi_2} Q_2 (L_i, L_j | x_i, x_j) \tag{3}$$

is the total energy relayed from the layout statistical model and the translational symmetry modules (Fig. 1). Ψ_1 and Ψ_2 are the neighbourhoods defined based on the short- and long- range edges (Sect. 2.2). $Q_1 (\cdot)$ is the prior for the plausible structural adjacencies, while $Q_2 (\cdot)$ is the regularizer for the translational symmetry of structures in the architectural scene at hand. The assigned label of a superpixel is mapped to all pixels sharing its basin.

In the following sections, we explain how the appearance and layout priors are established to be incorporated in our energy function for the TRW-S algorithm.

2.1 Appearance Cues

A well-known fact about visual perception is, it is evoked by appearance. Thus, our algorithm is launched by obtaining preliminary classification of the image superpixels that utilizes textural characteristics of the regions. We choose Random Forest (RF) as our classifier [2], which performs a recursive partitioning of the data based on an ensemble of decision trees. But, other efficient classifiers can be used instead.

Another critical choice is the space in which the feature vectors are embedded. We examine 2 spaces. Firstly, the vector $\mathbf{s}_i$ is comprised of the 128 SIFT descriptors [9], calculated densely over the image with a bin size of 8. Secondly, the vector $\mathbf{r}_i$ (4) and (5) is the distances to M predefined clusters, corresponding to M architectural structures. Each cluster consists of the SIFT feature vectors of the superpixels, belonging to a certain structure and acquired from the groundtruth data. The distance is calculated as the mean Euclidean norm between the SIFT vector of the superpixel and the k-nearest neighbours vectors in the cluster after removing the exact match. We preferred this distance over a centroidal one, because the clusters exhibit a high degree of scattering, due

to the high degree of appearance variation among instances of the same structure. Hence, the centroid would not be a proper representative of a cluster. we downsampled over-sized clusters to ensure a uniform prior for the RF.

$$\mathbf{r}_i = \left[r_i^1 \, r_i^2 \, \ldots\ldots \, r_i^M \right] \ . \tag{4}$$

$$r_i^j = \frac{\sum_{o=1}^{k} \left| \mathbf{s}_i - \mathbf{NN}_{ij}^o \right|}{k} \ . \tag{5}$$

$\mathbf{NN}_{ij}^o$ is the SIFT vector of the o-th nearest neighbour in cluster j with respect to data point i. And k is the count of neighbours.

In practice the later space was found to outperform the former. In our opinion, it introduced a higher level of semantics over the raw SIFT features, that achieved a substantial dimensionality reduction (from 128 to M features). The challenge for any dimensionality reduction algorithm is, not disturbing the position of a feature vector in its space, relative to label clusters. In the described space, we retain this relative position of the vector, by storing its distances to the clusters in the space, without the overhead of low-level SIFT details. In addition, this space transformation provided better characteristics for the training vectors, namely inter-separability and intra-compactness of the clusters. These characteristics are expected to boost, not only k-nn equivalents but also margin-maximizing hyperplane classifiers. However, further investigation is required to evaluate the proposed idea with other classifiers and clusters of various topologies. Similar approach of using a meta-feature vector can be found in [3]. The resulting segmentations are provided as input to the next phase. We also retain the classification probabilities $P\left(L_i|x_i\right)$ computed by the RF for each super-pixel to be used as datacosts in the TRW-S framework.

2.2 Layout Cues

In this module, we make use of 5 architectural principles, namely, spatial coherence, approximate structural location, structure ordering, recurring structural adjacencies, and translational symmetry. In our framework, these principles are expressed in the edge costs of the TRW-S graph. The edge costs are look-up tables giving the penalties for various combinations of labelings for the edge vertices. There are 2 types of edges: short-range and long-range.

Short-range Edges. They specify neighbours based on spatial proximity, and their edge costs used to establish $Q_1\left(\cdot\right)$ for the TRW-S function (3). Superpixels are connected by an edge if there is a common boundary between their encompassing basins. Hence, each superpixel is allowed a different number of neighbours. During the learning phase, we build a statistical model of the found adjacencies among structures. We argue that the familiar adjacencies is the most stable feature across different architectural scenes. For instance, a door structure can be seen adjacent to a wall, but never next to a sky structure.

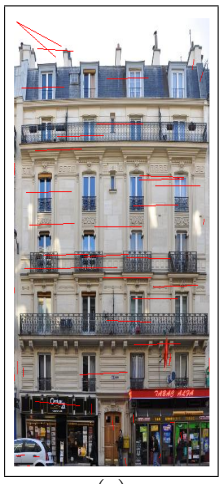
The edge costs are $M \times M$ matrices, where M is the number of architectural features encoding the costs for different combinations of labels for adjacent superpixels. We introduce the concept of *location-aware* edges, which entails different costs for edges in different zones of the facade. In POTTs model [1], the diagonal values are set to zero encourage neighbouring nodes get the same label. However, we utilize a non-POTTs model, in which the values on the diagonals of the cost matrices are non-zeros. Therefore, there is a penalty incurred even if nodes are given the same labeling. This penalty is dependent on the frequency by which the label has been seen in this zone of the image in the training samples. The frequencies of the labels with respect to locations are obtained through the following procedure. To account for image size variability, the groundtruth images are transformed to an approximate scale invariant space. This is done by subdividing each image into k horizontal and k vertical stripes of equal width, such that k^2 rectangular patches are formed. The corresponding patch is determined for each labeled pixel and the information is used to update the frequency of the label in the patch. The values are then normalized by dividing by the total pixel density within the patch to get probability $\mathbf{P}^m_{rc}$, such that $r, c \in \{1, 2, \dots k\}$.

To fill the upper and lower triangles of the cost matrices, we build a $2d$ histogram for the structural tangencies based on the same image subdivision, but this time for a pair of labels (instead of a single label) to encode a transition. The recorded frequencies in each patch, are normalized per structure to reflect the probability $\mathbf{P}^{ab}_{rc}$ that a pair of labels (a and b) exist in adjacency at this location, when a testing sample is introduced. $a, b \in \{M \times M\}$, such that $a \neq b$. Edges and their cost matrices are established in 2 directions corresponding to the directions for tangency: horizontal and vertical. For each structure instance in the ground truth, we record the structures to its east and south. We bypass the west and north directions because they are inverses of the included directions and would only require a transpose of the cost matrix. So, including them will redundantly duplicate the cost. The matrices are non-symmetrical. For instance, a roof structure is more frequently seen to the south of sky than to its north.

In this way, the edge cost matrices (Fig. 2) encode the architectural principles of, vertical and horizontal arrangement ordering of structures, in addition to locations and structural direct adjacencies. At inference time, if basins are tangent in both directions, we choose the direction of the common boundary with the longest length. We convert the probabilities to costs to build labeling penalty matrices, according to the Boltzmann distribution, $\mathbf{E}^m_{rc} = -\log\left(\mathbf{P}^m_{rc}\right)$ and $\mathbf{E}^{ab}_{rc} = -\log\left(\mathbf{P}^{ab}_{rc}\right) + \xi$. We add ξ, a constant to raise the range of values in the upper and lower triangles of the cost matrices over the diagonal values, to bias the optmization algorithm towards same labeling for the vertices of the edge. As such, spatial coherence is achieved while promoting the frequently encountered label in the training set, at this location. If the algorithm chooses to label the vertices differently, the most frequent adjacencies at this location are preferred.

Some practical adjustments were carried out, because the subdivision of the image is arbitrary and to prevent over-fitting to training data. We apply a Gaus-

sian smoothing filter on the frequency histograms of location and structural adjacency. In addition, Inf costs, resulting from zero frequency, are replaced by a relatively high value π, to discourage rather than eliminate the possibility of an assignment. Same goes for Inf values in the appearance datacost, as they are replaced by ρ.



(a)

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	21	464	464	21	464	464	464	464	464	21	21
2	464	2	21	464	21	464	21	464	21	21	21	21
3	464	464	7	464	464	464	464	464	464	464	464	464
4	464	464	464	444	464	464	464	464	464	464	464	464
5	464	464	464	464	5	464	464	464	464	464	464	464
6	464	464	464	464	464	444	464	464	464	464	464	464
7	464	20	464	464	464	464	5	464	464	464	464	464
8	464	464	464	464	464	464	464	444	464	464	464	464
9	464	21	464	464	464	464	464	464	5	21	464	464
10	464	21	464	464	21	464	464	464	464	4	464	464
11	464	464	464	464	464	464	464	464	464	464	444	464
12	464	464	464	464	464	464	464	464	464	464	464	6

(b)

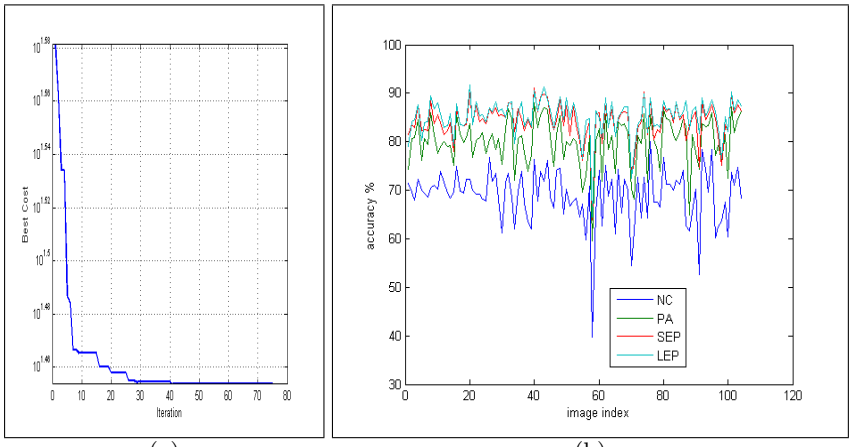
Fig. 2. (a) A sample of long-range edges shown in red. (b) A sample of short-range edge approximated cost matrix for the CMP dataset [14]. Structure 1 incurs the least cost, which signals that it is the most frequently encountered structure in this image patch. The most abundant transition is between structures 7 and 2. Structures 4, 6, 8, and 11 are never seen in this image patch during training. Values on the diagonal are in a lower range than the ones on the lower and upper triangles to promote same labeling.

Long-range Edges. These encode the translational symmetries found in the scene, used for building the $Q_2(\cdot)$ (3). To establish these symmetries we use the α-expansion graphcut algorithm [8], to assign a translation vector to each superpixel in the image. The ultimate goal is to establish a smoothness prior over distant instances of the same structure, in the TRW-S labeling optimization step. It is run separately for each type of putative structure resulting from the appearance classification phase. A Markov Random Field (MRF) is defined over all superpixels belonging to the structure and forming the nodes of the graph. The smoothness prior is based on neighbourhood Ω, detected between superpixels when their basins share a common boundary and belong to the same putative structure. Neighbourhoods are assigned a constant weight. The terminal nodes of the graph of the α-expansion algorithm constitute the labels and they are a

set of translational vectors. This set is constructed from the SIFT feature points of the image and their best matches. The matching score is calculated based on Euclidean norm in the SIFT space. The set of translational vectors is refined by preserving only the ones that exhibit a translation in either the x and y directions but not both. As such the long-range cliques promote the vertical and horizontal alignment of facade structures. The energy function E, to be minimized by the graphcut, is as follows:

$$E\left(Y\right) = \sum_{x_i} D_Y\left(y_i|x_i\right) + \mu \sum_{x_i} \sum_{x_j \in \Omega} F_Y\left(y_i, y_j|x_i, x_j\right) + \theta \cdot |Y_T| \ . \tag{6}$$

The unary term $D\left(\cdot\right)$ is the dissimilarity score between an examined superpixel x_i and the superpixel of the watershed basin, to which the destination belongs. The destination is obtained when applying translation y_i ($\in T$) on the examined superpixel. We constraint the translations to result in destinations being within image boundaries, but not necessarily belonging to the same structure as the source superpixel, to minimize the propagation of errors from the previous appearance-based stage. The pairwise term $F\left(\cdot\right)$ follows a POTTS model, in which a pair of neighbouring superpixels labeled differently, is penalized with a constant value. θ is a constant label cost that penalizes the assignment of x_i to new redundant labels. Redundancy in the sense that they can be replaced by one of the already utilized labels without drastically increasing the datacost Afterwards, the edges that will be relayed to the TRW-S algorithm are found by



(a) (b)

Fig. 3. (a) Semi-log scale plot of the cost against PSO iterations. (b) Accuracy plots for the images in ECP dataset when different options for IASC are activated.

applying on each superpixel its preferred translation vector in the specified, in addition to the reverse direction (a 180° rotated variant). In effect, this extends

the putative structures into a loci of points that complete their contained grids. An outcome of this phase is shown in Fig. 2.

2.3 Learning the Weight Parameters.

For learning the parameters of the energy functions, we use the Particle Swarm Optimization [11] (PSO) . A meta-heuristic technique, that relies on a user-specified range of values for the parameters. The algorithm initializes a swarm of vectors randomly. Each vector U_i holds values for the parameters and is named a particle. Iteratively, it updates the vectors based on their best previous position $U_{\text{i\_pbest}}$ and the best position in the swarm $U_{\text{global\_best}}$. The quality of the particle is evaluated based on a cost function. In all our experiments, the cost function is single objective.The position update rule for the i^{th} particle is

$$U_i = U_i + V_i \ . \tag{7}$$

The velocity V_i of the particle is given by,

$$V_i = \omega \times V_i + c_1 \times \text{rand}\,() \times (U_{\text{i\_pbest}} - U_i) + c_2 \times \text{rand}\,() \times (U_{\text{global\_best}} - U_i) \ . \tag{8}$$

The rule guarantees that the procedure yields non-increasing cost values in each iteration Fig. 3, thus leading to convergence. First, we use the PSO in learning the α-expansion parameters (θ and μ). In this case, the objective is minimizing the number of erroneous edges that link superpixels belonging to different genre of structures. In the second setting, it is used for optimizing β_1, β_2, ξ , π, and ρ in the TRW-S framework. The objective is minimizing the errors in the final labeling of the superpixels, when compared to ground truth data.

3 Evaluation

We follow the convention of related work, and document the results based on 5-fold cross validation and using pixel-based accuracy as the criterion for comparison. The training folds are used for constructing SIFT clusters of the structures, collecting the layout statistics and training the Random Forest. We test our model IASC (Integrated Appearance Structure Cues) on the *ECP-Monge* dataset [13] and the *CMP* dataset [14], and compare to the *state-of-the-art* results from the 3-layered approach [10], Spatial Pattern Templates (SPT) [14] and Auto-Context [6]. The ECP-Monge contains 104 images of facades in Hausmannian style. We use the corrected groundtruth [10]. The CMP is considered more challenging as it contains 378 samples from various (often difficult to model) styles. Because, we propose a multi-phase algorithm, we needed to separately examine each phase to understand its contribution to the final accuracy value. Table 1 summarizes the mean accuracies achieved by [10], [6] and [14] and IASC algorithm in various stages. We include the results of the commonly used POTTS model for spatial smoothness (PA), as a variant of our algorithm, and use the same datacosts of the IASC. We follow the naming conventions of the original

Table 1. Average accuracies on datasets. NC: No context (appearance only), AP: Aligned Pairs, APRT: Aligned Pairs Regular Triplets, SH: Structural Heuristics, PA: POTTS Adjacency, ST3: Auto-Context classified, PW3: POTTS Smoothed Auto-Context, SEP: Short-range Edges Prior, and LEP: Layout Edges Prior (short- and long- range).

	SPT [14]			3-layers [10]			Auto-Context [6]		IASC (our method)			
	(NC)	(AP)	(APRT)	(NC)	(PA)	(SH)	(ST3)	(PW3)	(NC)	(PA)	(SEP)	(LEP)
ECP	59.6	79.0	84.2	82.6	85.1	84.2	90.8	91.4	68.9	79.9	86.3	87.8
CMP	33.2	54.3	60.3	-	-	-	66.2	68.1	41.4	55.5	60.3	64.4

papers [14, 10, 6] in reporting results. Per-image accuracies are shown in Fig. 3, for the different factors affecting the performance of our model. In Fig. 4, we display results of a selection of samples. We can conclude from experiments, for IASC, each phase consistently improved accuracy over the preceding one. Despite our efforts to minimize the propagation of errors, across the system modules, it is evident that appearance classification failures remain a limiting factor for subsequent improvements. It is evident for [10], the incorporation of the structural heuristics (such as: the existence of a running balcony on the second and fifth floor) degraded the accuracy of their smoothed appearance classifications. As for [14], the fact that their neighbourhoods of pairs and triplets were based on a manually assigned threshold was a severe limitation. The reported result for ECP-Monge in [6] is based on 7 classes of structures, whereas we include the result using the updated groundtruth which added the chimney structure. In IASC, we record one of the highest accuracy net gains when incorporating layout cues in the problem of facade parsing, even when starting with severely damaged results based on appearance. This is attributed to the generalization ability of our optimization function that relies only on persistent architectural guidelines without being style specific.

We use the Davies Bouldin (DB) index [4] to shed light on the characteristics of the proposed feature space of distance-to-cluster, against the raw SIFT feature space. The clustering is predefined from the groundtruth and we normalized the 2 spaces. It was found that the proposed space transformation increased both separability and compactness of the clusters, thus, favorably lowering the average DB on the training folds from 8.4616 to 1.4497. As for classification accuracy, raw SIFT vectors achieved 63.3% on ECP-Monge in the No Context setting. For the distance vectors, the figure was 68.9%. In both settings we use the PSO to learn the parameters, the number of iterations was set to 75. The swarm size was 10 when optimizing the parameters for finding long-range edges and 40 for the TRW-S function. The parameters ranges were based upon our observations during experiments, but we provided a much wider range to lower the risk of a local minimum. In evaluating the objective functions, 10 samples were selected randomly for each dataset. The objective function yields the highest calculated cost based on the 10 samples.

4 Conclusion

We present an algorithm for handling semantic segmentation of architectural scenes. The algorithm relies on the output of a Random Forest classifier on SIFT-based meta-feature vectors. We carry out a feature space transformation from raw SIFT to distance-to-cluster vectors. Also, we incorporate layout principles in the form of labeling costs for superpixel long-range cliques resulting from translation vectors, detected by α-expansion. Other labeling costs are based on location and structural adjacencies, defined on short range neighbourhoods. We report competitive results. We believe our method offers significant advantages over competitors in terms of algorithm elegance. The priors are automatically learned from training samples and its weight parameters are deduced via the single objective PSO algorithm. At inference time, the labeling is efficiently optimized using the TRW-S algorithm, while including no heuristics or manually determined thresholds. Our future work is intended towards boosting the accuracy figures, by plugging in the state-of-the-art Convolutional Neural Networks in the appearance module and relaying its resulting posteriors to our layout optimization function.

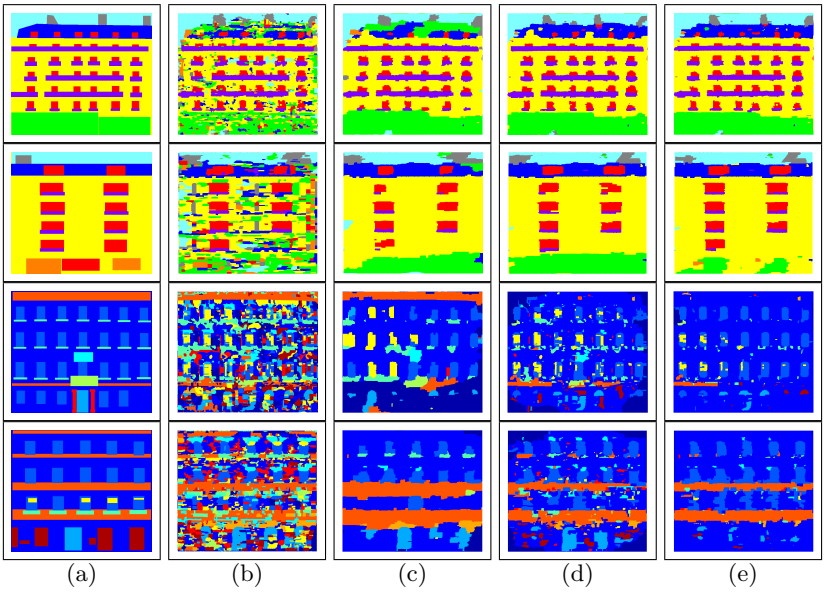


Fig. 4. Sample outcomes in tabular format. Row (1) ECP-Monge sample with accuracy 91.72%; (2) ECP-Monge sample with accuracy 83.18%; (3) CMP sample with accuracy 74.21%; (4) CMP sample with accuracy 72.00%. Column (a) Ground truth; results of (b)NC; (c) PA; (d) SEP; (e) LEP.

References

[1] Y. Boykov, O. Veksler, and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, Nov. 2001.

[2] L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[3] K. Dang and J. Yuan. Location Constrained Pixel Classifiers for Image Parsing with Regular Spatial Layout. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

[4] D. L. Davies and D. W. Bouldin. A Cluster Separation Measure. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1(2):224–227, Feb. 1979.

[5] P. Dollar, Z. Tu, P. Perona, and S. Belongie. Integral Channel Features. In *Proc. BMVC*, pages 91.1–91.11, 2009.

[6] V. Jampani, R. Gadde, and P. V. Gehler. Efficient 2D and 3D Facade Segmentation Using Auto-context. In *Proceedings of the 2015 IEEE Winter Conference on Applications of Computer Vision*, WACV '15, pages 1038–1045, Washington, DC, USA, 2015. IEEE Computer Society.

[7] V. Kolmogorov. Convergent Tree-Reweighted Message Passing for Energy Minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568–1583, Oct. 2006.

[8] V. Kolmogorov and R. Zabih. What Energy Functions Can Be Minimized via Graph Cuts? In *Proceedings of the 7th European Conference on Computer Vision-Part III*, ECCV '02, pages 65–81, London, UK, UK, 2002. Springer-Verlag.

[9] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, 60(2):91–110, Nov. 2004.

[10] A. Martinović, M. Mathias, J. Weissenberg, and L. Van Gool. A Three-Layered Approach to Facade Parsing. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision – ECCV 2012*, volume 7578 of *Lecture Notes in Computer Science*, pages 416–429. Springer, 2012.

[11] R. Poli, J. Kennedy, and T. Blackwell. Particle swarm optimization. *Swarm Intelligence*, 1(1):33–57, 2007.

[12] R. Socher, C. C. Lin, A. Y. Ng, and C. D. Manning. Parsing Natural Scenes and Natural Language with Recursive Neural Networks. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, 2011.

[13] O. Teboul, I. Kokkinos, L. Simon, P. Koutsourakis, and N. Paragios. Shape grammar parsing via Reinforcement Learning. In *CVPR*, pages 2273–2280. IEEE Computer Society, 2011.

[14] R. Tyleček and R. Šára. *Spatial Pattern Templates for Recognition of Objects with Regular Structure*, chapter Spatial Pa, pages 364–374. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[15] L. Vincent and P. Soille. Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(6):583–598, June 1991.