Edge Map Response of Dilated and Reconstructed Classical Filters

Ciprian-Constantin Orhei, Victor Bogdan and Cosmin Bonchis

September 7, 2020

Edge map response of dilated and reconstructed classical filters

Ciprian ORHEI
Politehnica University of Timisoara,
Timişoara, RO-300223
Romania.
ciprian.orhei@cm.upt.ro

Victor BOGDAN
West University of Timişoara
Bd. V. Pârvan 4
Timişoara, RO-300223
Romania.
victor.bogdan97@e-uvt.ro

Cosmin BONCHIŞ
West University of Timişoara and
The eAustria Research Institute,
Bd. V. Pârvan 4, 045B,
Timişoara, RO-300223
Romania.
cosmin.bonchis@e-uvt.ro

Abstract

Edge detection is a basic and fundamental feature in image processing domain. Dilation of edge filters kernels has proven to bring benefits for the edge detection operation by permitting to filter out noise and to take in consideration a bigger region of the image when processing. Numerous techniques were used in the past for finding edge features, one of the most common used being finding features in lower level scale of the image pyramid. Now, naturally, we want to investigate if our dilating of the filter kernels bring similar benefits as finding edges in a lower scale pyramid level.

Keywords

Dilated filters, Edge detection operator, Edge detection, Canny algorithm, Pyramid Level feature extraction.

I. Introduction

An edge in a image is one of the most basic features and has been intensively researched in the Computer Vision universe. A variety of mathematical methods have been used to identify points in which the image brightness changes sharply or has discontinuities. This is a fundamental tool in image processing, image analysis, machine vision and computer vision, particularly in the areas of feature detection and feature extraction.

Standard edge detection filters are built for highlighting intensity boundary changes in the near neighborhood image regions. The most frequently used methods can be found in: [Prewitt, 1970], [Sobel and Feldman, 1973], [Scharr, 2000]. Those filters are used in many computer vision algorithms which rely on edge detection for applications such as face recognition, target recognition, obstacle detection, image compression and many others.

In [Adelson et al., 1984] are presented two methods of edge detection over different pyramid levels. In the first approach the target patterns are constructed at several high pyramid levels, and each is convolved with the original image. In the second approach, a single copy of the target is convolved with copies of the image reduced in scale, see Figure 1. This is similar with what we want to present in this paper, by considering the target pattern as the edge detection operator. But even if we use a different pyramid level to determine them, ultimately we would like to bring them back in original level so we can use the results.

In this paper, we will analyze and compare the dilation of filters defined in the previous work [Bogdan et al., 2020] and the reconstruction approach for all considered filters. We try here the experimental proof of the hypothesis that dilating a 3x3 filter with a factor of 1 is similar with applying the same filter in the immediately lower scale pyramid level. This hypothesis stands because in both cases the region we take in consideration to find edges isn't anymore an 3x3 matrix but a 5x5 matrix. We can continue this logic of dilating with a factor of two, which is similar with applying the operation at two levels difference. Feature extraction in lower pyramid scale level is a common practice in the domain because of the benefits of lower computation resources needed and the loss of details.

In section II, we will present the necessary information and background for our experiments and simulations. In section III,we show the steps used to bring the edge maps that we found in lower scale pyramid level back to initial level and we highlight first visual results using a controlled set of images. Furthermore, we present the results on a natural image data set using standard filters (in section IV) and by using Canny algorithm (in section V). Also, in both sections we present the equivalence of dilated filters and the reconstructed lower pyramid level result in details.

II. Preliminaries

Dilated filters

In a previous work [Bogdan et al., 2020], we define dilation of a filter as an expansion of the original filter by a factor. By dilating the kernels, we propose to increase the distance between the pixels, distance which influences the result of the

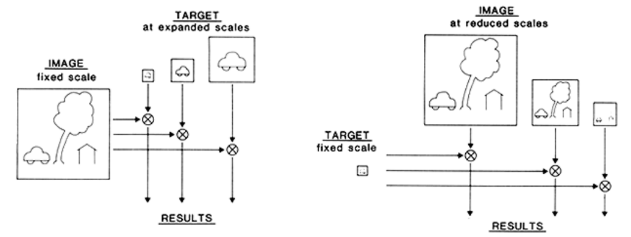

Fig. 1. Methods of working in different scales [Adelson et al., 1984].

convolution. This expansion induces the possibility of finding stronger intensity changes in the image on a bigger region of interest. When we dilate the kernels, we are filling the newly added positions with 0s. In order to highlight our definition on a filter,we will use the Sobel filter (Figure 2) from [Sobel and Feldman, 1973] and the dilation in Figure 3 and Figure 4.

$$\begin{bmatrix}1 & 0 & -1\\ 2 & 0 & -2\\ 1 & 0 & -1\end{bmatrix}\begin{bmatrix}1 & 2 & 1\\ 0 & 0 & 0\\ -1 & -2 & -1\end{bmatrix}$$

Fig. 2. Sobel Gx and Gy kernels

$$\begin{bmatrix}1 & 0 & 0 & 0 & -1\\ 0 & 0 & 0 & 0 & 0\\ 2 & 0 & 0 & 0 & -2\\ 0 & 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 0 & -1\end{bmatrix}\begin{bmatrix}1 & 0 & 2 & 0 & 1\\ 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\\ -1 & 0 & -2 & 0 & -1\end{bmatrix}$$

Fig. 3. Sobel 5×5 dilated Gx and Gy filterss

$$\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & -1\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 2 & 0 & 0 & 0 & 0 & 0 & -2\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 0 & 0 & 0 & -1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 2 & 0 & 0 & 1\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\\ -1 & 0 & 0 & -2 & 0 & 0 & -1\end{bmatrix}$$

Fig. 4. Sobel 7×7 dilated Gx and Gy filters

In the visual and statistical comparisons, we observe that dilating the filters, rather than extending them, helps to find more edge pixels than the standard filters. Another benefit worth mentioning of dilating is the fact that the number of operations doesn't increase with the dilating factor.

Pyramid Level Operations

From [Adelson et al., 1984], we can find operations regarding scaling between levels of the image pyramids. In Equation 1 the REDUCE operation is presented and in Equation 2 we can find the EXPAND operation, where G_l represents the image at level l and $G_{l,k}$ represents the image obtained by expanding G_l k times; $w(m,n)$ is the weighting function that is used for the REDUCE operation.

$$G_l(i,j) = \sum_m \sum_n w(m,n) G_{l-1}(i,j)(2i+m, 2j+n) \quad (1)$$

$$G_{l,k}(i,j) = 4 \sum_m \sum_n G_{l,k-1}(\frac{2i+m}{2}, \frac{2j+n}{2}) \quad (2)$$

The $Laplacianpyramid$ or bandpass pyramid, presented in Equation 3, is a bypass of the image transformation through scales of the pyramid and it holds the details(edges) that we lose in the REDUCTION process. In order to be able to RECONSTRUCT the image in the future, we preserve the details in the LAPLACE PYRAMID. Image reconstruction is presented in Equation 4.

$$L_l = G_l - EXPAND[G_{l+1}] \quad (3)$$

$$RECONSTRUCTED[G_l] = L_l + EXPAND[G_l] \quad (4)$$

Gradient operators

In this paper, we will use the following first order derivative operators for our analysis: [Prewitt, 1970], [Kirsch, 1971], [Sobel and Feldman, 1973], [Scharr, 2000], [Kawalec-Latała, 2014], [Kroon, 2009], details for the gradient for x axis only can be seen in Figure 5.

$$\underset{\text{Prewitt}}{\begin{bmatrix}1 & 0 & -1\\ 1 & 0 & -1\\ 1 & 0 & -1\end{bmatrix}} \quad \underset{\text{Kirsch}}{\begin{bmatrix}5 & -3 & -3\\ 5 & 0 & -3\\ 5 & -3 & -3\end{bmatrix}} \quad \underset{\text{Sobel}}{\begin{bmatrix}1 & 0 & -1\\ 2 & 0 & -2\\ 1 & 0 & -1\end{bmatrix}}$$

$$\underset{\text{Kayyali}}{\begin{bmatrix}6 & 0 & -6\\ 0 & 0 & 0\\ -6 & 0 & 6\end{bmatrix}} \quad \underset{\text{Scharr}}{\begin{bmatrix}3 & 0 & -3\\ 10 & 0 & -10\\ 3 & 0 & -3\end{bmatrix}} \quad \underset{\text{Kroon}}{\begin{bmatrix}17 & 0 & -17\\ 61 & 0 & -61\\ 17 & 0 & -17\end{bmatrix}}$$

Fig. 5. Gx kernels

We will consider in the following the standard equation where the gradient components, Gx and Gy, are used to define the gradient magnitude $|G|$ in Equation 5.

$$|G| = \sqrt{Gx^2 + Gy^2}. \quad (5)$$

From [Woods, 2011], we use the following steps to convolve filters with a source image in order to obtain the edge map:

Step 1 Convert the image to gray-scale.

Step 2 Reduce the noise in the source image by applying the Gaussian filter, in order to obtain smoother values.

Step 3 Applying the filters by convolving the gray-scale image with their kernels on the x and y axes and then applying Equation 5).

Step 4 Each pixel, which has an intensity value higher or equal to a *threshold*, will have its value set to $MaxValue$ (e.g. 255), else to 0, therefore the edges will be represented by the white pixels.

Step 5 Thinning the edges to one pixel width.

Canny edge operator

The Canny edge detection algorithm [Canny, 1986] which is a widely known and used edge detection algorithms, has the following summarized steps:

Step 1 Convert the image to gray-scale.

Step 2 Reduce the noise in the source image by applying the Gaussian filter, in order to obtain smooth values.

Step 3 Applying the filters by convolving the gray-scale image with their kernels on the x and y axes.

Step 4 Non-maximum suppression, for edge thinning of the obtained results.

Step 5 Edge tracking by hysteresis using double threshold.

From [Xu et al., 2011], we were inspired to process the threshold using he maximum pixel intensity in the input image and applying the formula from Equation 6. T_h is the upper threshold, T_l is the lower threshold and $max(input)$ is the maximum pixel intensity in the input image. From our experiments, the best results were obtained with the fixed weights values $w_h = 0.7$ and $w_l = 0.3$.

$$\begin{aligned} T_h &= max(input) \times w_h \\ T_l &= T_h \times w_l \end{aligned} \tag{6}$$

Benchmarking the edge operators

We use BSDS500 [Arbelaez et al., 2011] for highlighting the evaluation results. This dataset contains natural images that have been manually segmented, that serve as ground truth for the benchmark. The benchmark [Arbelaez et al., 2011] uses 500 test images, which are split in 3 different sets, each having at least 5 human segmented boundary ground-truth images. For evaluating the images generated from algorithms to the ground truth images, the Corresponding Pixel Metric (CPM) algorithm [Prieto and Allen, 2003] is used. This metric is reliable for correlating similarities with a small localization error in the detected edges. The metric first finds an optimal matching of the pixels between the edge images and then estimate the error produced by this matching.

$$P = \frac{TP}{TP + FP}. \tag{7}$$

$$R = \frac{TP}{TP + FN}. \tag{8}$$

$$F - measure = \frac{2 * TP}{2 * TP + FP + FN}. \tag{9}$$

For each image, we compute the following metrics, defined in [Sasaki, 2007]: Precision (P), Recall (R) and F-measure or F1-score (F1). For computing the metrics, we should consider: TP (True Positive) represents the number of matched edge pixel; FP (False Positive) the number of edge pixels which are incorrectly highlighted; FN (False Negative) the number of pixel that have not been detected. Precision, Equation 7, represents the probability of a resulting edge pixel is a true edge pixel. Recall, Equation 8, represents the probability that a true edge pixel is detected, where the two quantities are used to compute *F-measure* (F1-score) by applying the Equation 9.

III. Expanding edge maps

In this section, we will present the steps we used to reconstruct an edge map produced in a lower scale level pyramid, considering the original image being level zero. For a better illustration of the results we have used five synthetic grey-scale images presented in Figure 6, similar to the work we found in [Guizhen et al., 2007].

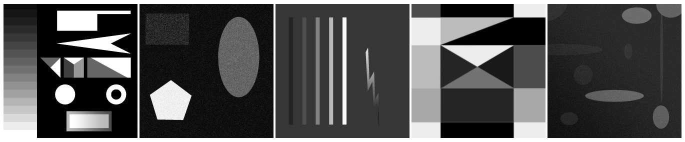

Fig. 6. Synthetic grey-scale test images

Applying the edge detection algorithm from [Canny, 1986] using the edge operator [Sobel and Feldman, 1973] with a dilation factor of 1, we obtain the results presented in Figure 7. And we also apply the same edge detector algorithm on pyramid level one but without any dilation, results are presented in Figure 7.

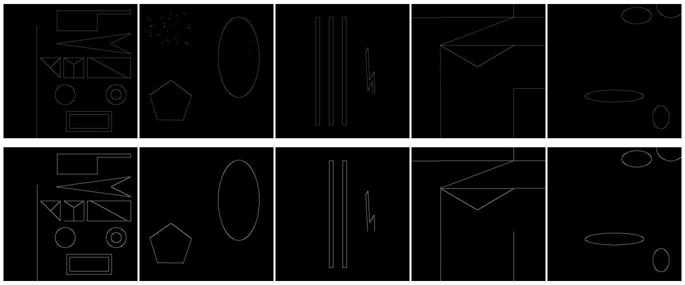

Fig. 7. Rows: Canny using Sobel dilated 5x5 results on synthetic images on L0, Canny using Sobel 3x3 on synthetic images on L1

The first experiment we did was to REDUCE the results of Canny using Sobel dilated 5x5 filter from level zero to level one and compare them with the results of Canny using Sobel 3x3 filters on level one. In order to observe the actual differences better, we applied a threshold on the reduced image of value 60 so we can eliminate the false edge pixels. We can observe in Figure 8 the result of REDUCE operation. In Figure 9, we present the edge pixels found only by applying Canny with Sobel 3x3 and the edge pixels found only by applying Canny with Sobel Dilated 5x5. In calculating of the difference between the edge maps, we considered an offset of one pixel in any direction, offset that is caused by the reduction operation.

Our intuition, that dilation in pyramid level zero is equivalent with applying the original operator in pyramid level one, was confirmed by the difference in the results which can be noticed in Figure 9. By observing Figure 9, we can see the extra edges that the dilated filter actually found.

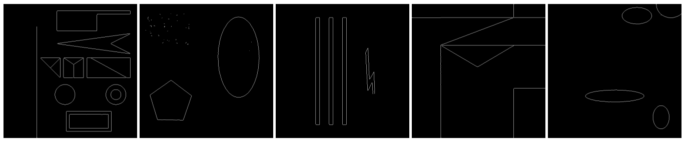

Fig. 8. Canny using Sobel dilated 5x5 on synthetic images on L0 reduced to L1

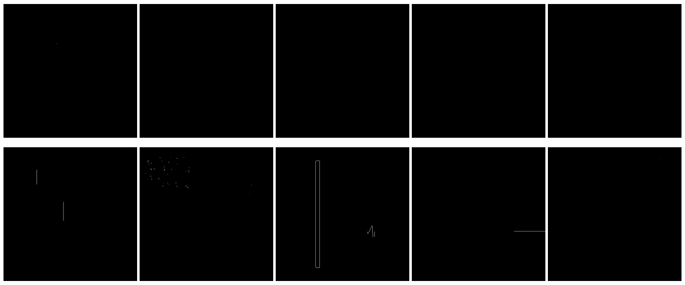

Fig. 9. Rows: edge pixels found only by Canny using Sobel 3x3 on L1; edge pixels found only by Canny using Sobel dilated 5x5 on L1

There are many variants in literature that are using different image levels for feature detection and reconstruction to initial level. For the purpose of our comparison, we chose three different variants of expanding the edge map:

V1 Expanding the edge image from level 1 to level 0.
V2 Reconstructing the edge image from level 1 to level 0.
V3 Intersect the expanded edge maps from level 1 with edge map from level 0.

In the following, we detail the enumerated variants of transforming an edge map from level one to level zero pyramid.

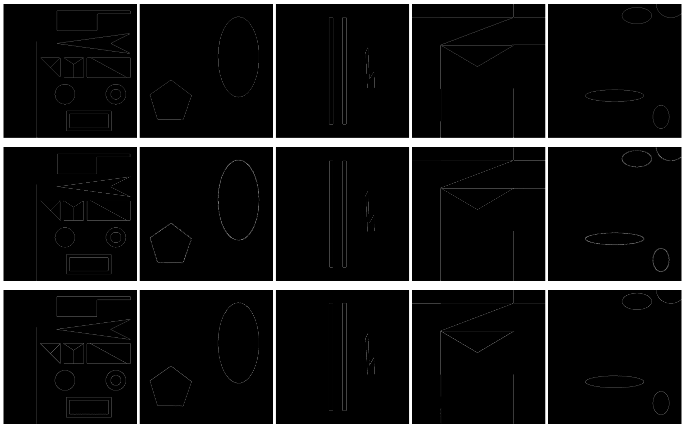

Fig. 10. Rows: Edge map pixels expanded from L1 using Variant 1; Edge map pixels expanded from L1 using Variant 2; Edge map pixels expanded from L1 using Variant 3

Variant 1

In this variant, we consider just expanding starting from an edge map in the first pyramid level, approach described in the following steps:

Step 1 EXPAND the edge map using Equation 2.
Step 2 Transform each edge map pixel value to 255, we can do this step because it is not significant at this point if it is an strong or weak edge.
Step 3 Apply the thinning algorithm presented in [Guo and Hall, 1989] to remove the excess edge points caused by the expansion.

The results of this variant of reconstructing the Canny resulted edge map to level zero on the synthetic images are presented in Figure 10.

Variant 2

Similar to [Lai et al., 2017], where the images are reconstructed for a Deep Laplace Network, we consider the following steps to expand the result of Canny Sobel 3x3 on level one to level zero:

Step 1 EXPAND the image using Equation 2.
Step 2 Calculate the Laplace Pyramid for level zero using Equation 3.
Step 3 Intersect the expanded image and Laplace Pyramid image. We do the intersection instead of an reunion of pixels to avoid adding edge pixels that were discovered by the Laplace pyramid but not by Canny on level one.
Step 4 Apply the thinning algorithm [Guo and Hall, 1989] because of the expansion we tend to have 4 pixels width edges and we desire one pixel edges.

The results of this variant of reconstructing the Canny resulted edge map on level zero on the synthetic images are presented in Figure 10.

Variant 3

Similar to [Wang et al., 2018] and [Liu et al., 2019], where the edge map is fused by averaging the expansions from different levels, we consider the following steps to transform the result of Canny Sobel 3x3 on level one to level zero:

Step 1 EXPAND the edge map using Equation 2.
Step 2 Calculate the edge map on level zero.
Step 3 Intersect the expanded edge map from level one and edge map on level zero . We do the intersection instead of an reunion of pixels to avoid adding edge pixels that were discovered by Canny done on level zero and not by Canny on level one.

The results of this variant of reconstructing the Canny resulted edge map on level zero on the synthetic images are presented in Figure 10.

The presented variants are recursively repeated for each level we want to expand from. In Figure 11 we present the results for the three variants when expanding from level two to level zero of the image pyramid and using 7×7 dilated filter in level 0 for comparison. We can notice from this comparison that the dilation factor n of a filter and its application on an image is equivalent to applying the standard filter directly on the pyramid level L_n of the same image.

As we expected looking over the results from this section, we can observe that we obtain similar results from reconstructing the edge map or dilating the filters. Of course, we have applied this in a control data set and in the next section we want to use the variants we exposed in a bigger natural set of images.

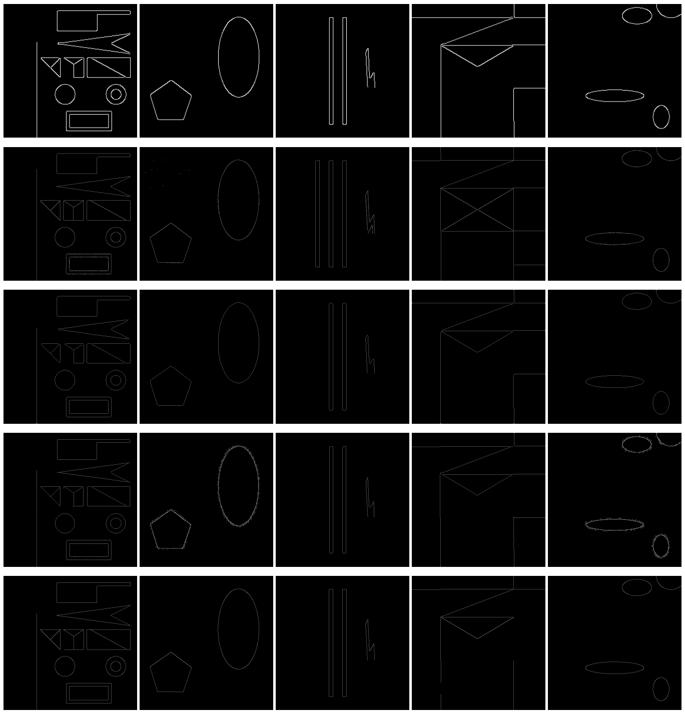

Fig. 11. Rows: Canny using Sobel dilated 7x7 on synthetic images on L0; Canny using Sobel 3x3 on synthetic images on L2; Edge map pixels expanded from L2 to L0 using Variant 1; Edge map pixels expanded from L2 to L0 using Variant 2; Edge map pixels expanded from L2 to L0 using Variant 3

IV. Filter convolution results

The following section consists of comparisons between the results of convolving an image with the dilated filters and the result of expanding a convolution on a lower scale level of pyramid. We consider the filters [Prewitt, 1970], [Kirsch, 1971], [Sobel and Feldman, 1973], [Scharr, 2000], [Kawalec-Latała, 2014], [Kroon, 2009] presented in Section II. The results of filter edge detection using Equation 5 are shown in Figure 12.

In Figure 13 and Figure 14, we can see the visual results of the three reconstruction variants we presented in Section IV.

Visual comparison of the results have shown that the results are similar between the expanded variants results and the dilatation results. This maintenance of the trend in the results from section III is encouraging. In order to understand better the results, we are going to evaluate the results by using the BSDS500 benchmark from [Arbelaez et al., 2011], for details see Section II. We have used as data set for this evaluation all 500 images that the Benchmarks offers. The results of the benchmark process can be observed in Table I.

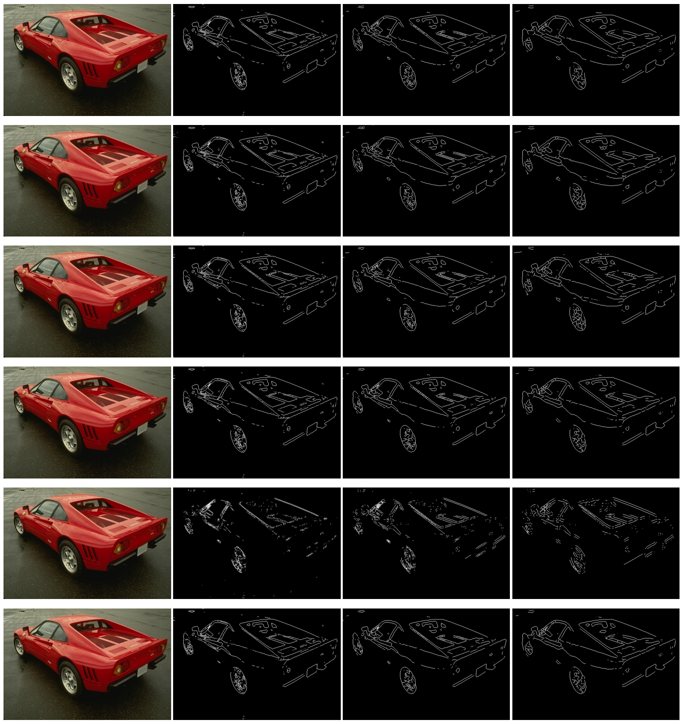

Fig. 12. Edge Filter Operator results. Columns are: original image, 3x3 kernel filter, 5x5 dilated filter, 7x7 dilated filter. Rows: Sobel, Prewitt, Kirsch, Scharr, Kayyali, Kroon

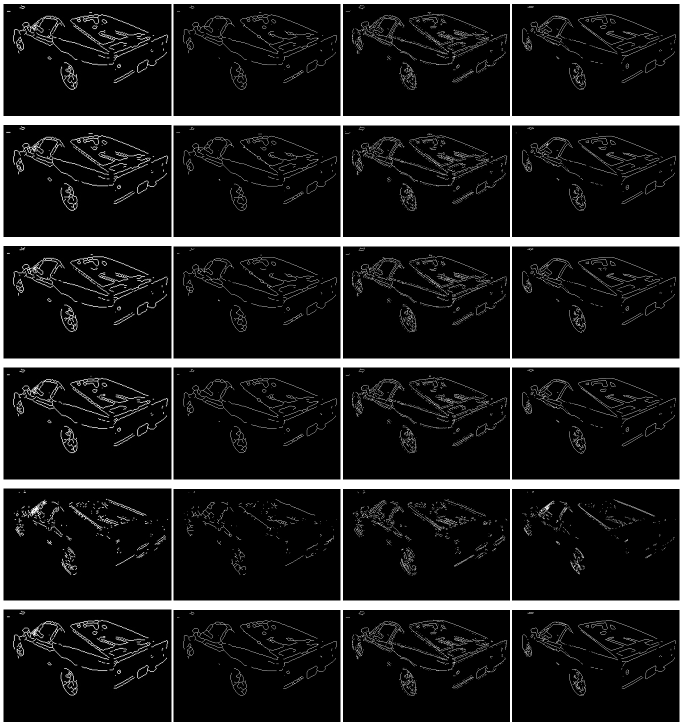

Fig. 13. Edge Filter Operator reconstructed from L1. Columns are: 3x3 kernel on L1, reconstructed using Variant 1, reconstructed using Variant 2, reconstructed using Variant 3. Rows: Sobel, Prewitt, Kirsch, Scharr, Kayyali, Kroon

From the statistical data, we can observe that the metrics

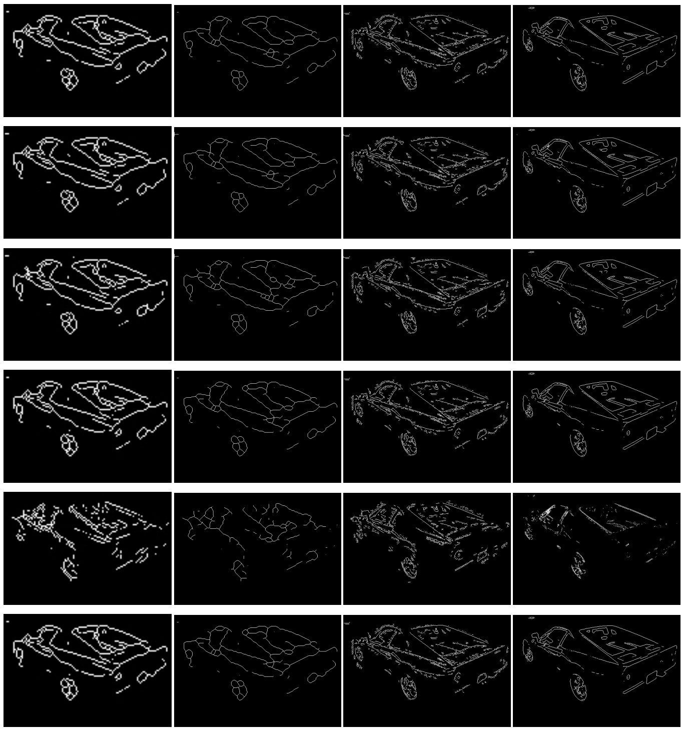

Fig. 14. Edge Filter Operator reconstructed from L2. Columns are: 3x3 kernel on L2, reconstructed using Variant 1, reconstructed using Variant 2, reconstructed using Variant 3. Rows: Sobel, Prewitt, Kirsch, Scharr, Kayyali, Kroon

Variant		3x3	Dilated 5	V1 L1	V2 L1	V3 L1	Dilated 7	V1 L2	V2 L2	V3 L2
Sobel	R	**0.688**	0.663	0.571	0.687	0.595	0.659	0.425	0.642	0.595
	P	0.439	0.496	0.566	0.488	0.540	0.518	**0.584**	0.435	0.540
	F1	0.536	0.568	0.568	0.571	0.566	**0.580**	0.492	0.519	0.566
Prewitt	R	**0.687**	0.661	0.571	0.685	0.593	0.662	0.426	0.641	0.593
	P	0.441	0.500	0.567	0.490	0.544	0.518	**0.583**	0.435	0.544
	F1	0.538	0.569	0.569	0.572	0.567	**0.581**	0.492	0.518	0.567
Kirsch	R	0.687	0.658	0.574	**0.688**	0.594	0.643	0.432	0.644	0.594
	P	0.441	0.500	0.568	0.491	0.544	0.529	**0.587**	0.437	0.544
	F1	0.537	0.568	0.571	0.573	0.568	**0.580**	0.497	0.521	0.568
Scharr	R	0.686	0.660	0.571	**0.688**	0.595	0.657	0.425	0.643	0.595
	P	0.437	0.493	0.565	0.487	0.538	0.516	**0.584**	0.435	0.538
	F1	0.534	0.565	0.568	0.570	0.565	**0.578**	0.492	0.519	0.565
Kayyali	R	0.378	0.357	0.300	0.476	0.280	0.393	0.223	**0.502**	0.280
	P	0.422	0.472	0.517	0.444	0.501	0.498	**0.544**	0.407	0.501
	F1	0.399	0.407	0.380	**0.460**	0.359	0.439	0.316	0.449	0.359
Kroon	R	0.686	0.660	0.571	**0.688**	0.595	0.656	0.424	0.643	0.595
	P	0.437	0.493	0.565	0.487	0.538	0.515	**0.584**	0.434	0.537
	F1	0.534	0.564	0.568	0.570	0.565	**0.577**	0.492	0.519	0.565

TABLE I
EDGE FILTERS RESULTS

are similar between the dilated edge operators and the reconstructed version from the lower levels. The most important results that we can see from Table I is that the dilatation results have similar by using less computational resources. There are also another two observations which we can notice from those numbers. First, by looking to the F1 scores, we can observe that for different operators we can obtain equivalence with different variant expansions (e.g. F1 dilated 5x5 Sobel is similar with V1 L1 Sobel expansion, but with V3 L1 for Kirsch expansion). Second, it seems that in most of the cases the precision is better in the reconstructed lower scale level than the direct dilation application, but it is explainable due to the noise reduction of the pyramid levels.

V. CANNY SIMULATION RESULTS

Now, we present the comparisons between the results of the Canny algorithm using filters both presented in II and the result of expanding a on a lower level of pyramid presented in section IV. Visual results can be observed in Figure 15 and the expansion results in Figure 16 and Figure 17. We can observe the expected equivalence from all visual comparison with slight differences in the expansion variants.

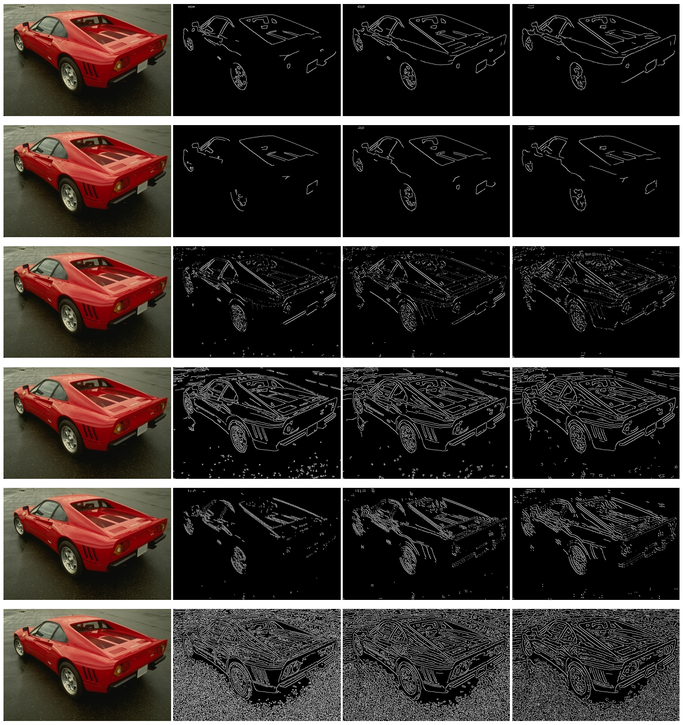

Fig. 15. Canny Edge results. Columns are: original image, 3x3 kernel filter, 5x5 dilated filter, 7x7 dilated filter. Rows: Sobel, Prewitt, Kirsch, Scharr, Kayyali, Kroon

Because of the nature of the Canny algorithm, it is predictable that applying it on a lower scale pyramid level has benefits. Images on this levels tend to lose details that can be considered noise when we talk about edges or contour detection. As we can see from the visual comparison the result of the algorithm applied on a different level than the original tend to highlight more significant edges. But, we can observe a continuity related to the results trend between the edge maps obtained in pyramid level zero.

Similar to Section IV, we compared statically the Canny results and Table II, we can observe the benchmark output for all the operators using dilation or lower scale pyramid level expansions.

If we look upon the results from Table II, we can observe that the precision is better in case of reconstructed variants that translates in a better hit rate of edge pixels produced by the algorithm. But similar we can observe a better recall when we talk about dilated filters that translate into a better accuracy in detecting edge pixels. This shows similar F1 scores

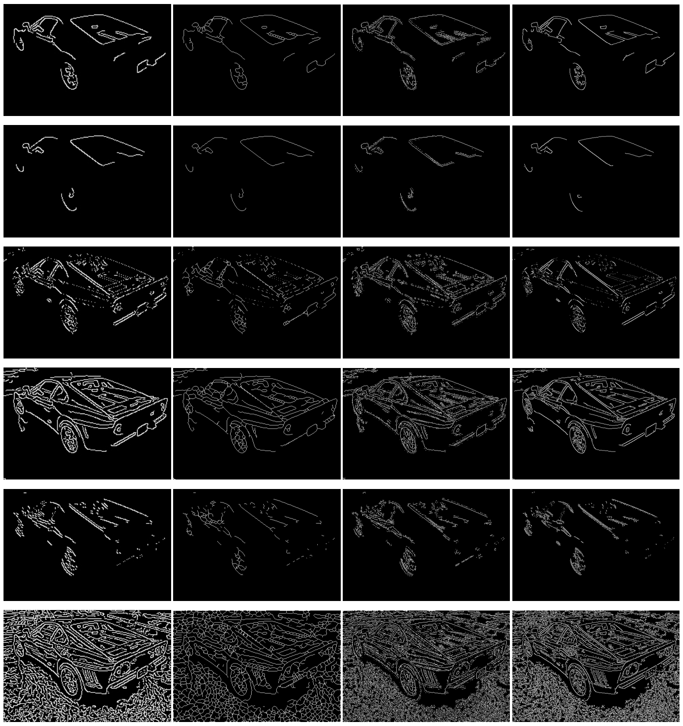

Fig. 16. Canny Edge reconstructed from L1. Columns are: 3x3 kernel on L1, reconstructed using Variant 1, reconstructed using Variant 2, reconstructed using Variant 3. Rows: Sobel, Prewitt, Kirsch, Scharr, Kayyali, Kroon

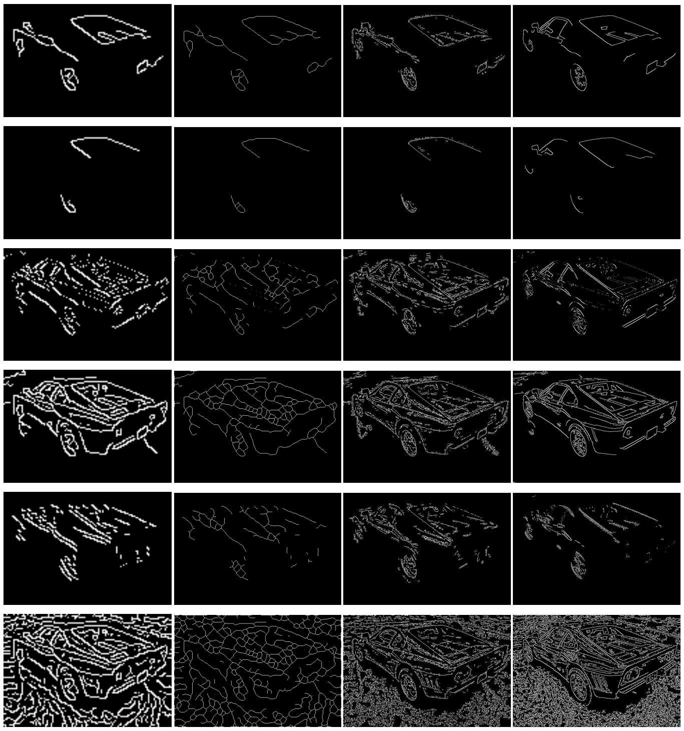

Fig. 17. Canny Edge reconstructed from L2. Columns are: 3x3 kernel on L2, reconstructed using Variant 1, reconstructed using Variant 2, reconstructed using Variant 3. Rows: Sobel, Prewitt, Kirsch, Scharr, Kayyali, Kroon

Variant		3x3	Dilated 5	V1 L1	V2 L1	V3 L1	Dilated 7	V1 L2	V2 L2	V3 L2
Sobel	R	0.467	0.596	0.320	0.380	0.336	**0.599**	0.242	0.336	0.336
	P	0.535	0.537	0.709	0.639	0.689	0.556	**0.771**	0.563	0.689
	F1	0.499	0.565	0.441	0.477	0.452	**0.577**	0.369	0.421	0.452
Prewitt	R	0.281	0.411	0.170	0.202	0.173	0.415	0.124	0.166	0.173
	P	0.616	0.620	0.785	0.732	0.781	0.643	**0.853**	0.635	0.781
	F1	0.386	0.494	0.280	0.316	0.283	**0.504**	0.216	0.263	0.283
Kirsch	R	0.700	0.731	0.516	0.745	0.527	**0.795**	0.353	0.674	0.527
	P	0.372	0.383	0.464	0.408	0.455	0.377	**0.519**	0.394	0.455
	F1	0.486	0.503	0.489	**0.527**	0.489	0.512	0.420	0.497	0.489
Scharr	R	0.940	0.957	0.833	0.912	0.899	**0.955**	0.613	0.840	0.899
	P	0.265	0.282	0.399	0.321	0.333	0.300	**0.472**	0.332	0.333
	F1	0.413	0.436	**0.539**	0.475	0.486	0.457	0.533	0.476	0.486
Kayyali	R	0.459	0.715	0.246	0.381	0.277	**0.788**	0.154	0.339	0.277
	P	0.412	0.377	0.534	0.469	0.515	0.372	**0.593**	0.452	0.515
	F1	0.434	0.494	0.337	0.421	0.360	**0.505**	0.244	0.387	0.360
Kroon	R	**0.996**	0.994	0.929	0.980	0.986	0.989	0.679	0.907	0.986
	P	0.179	0.203	0.268	0.217	0.209	0.215	**0.335**	0.243	0.209
	F1	0.304	0.337	0.416	0.356	0.345	0.354	**0.448**	0.383	0.345

TABLE II
CANNY RESULTS

obtained by the reconstructed variants and dilated ones. If we look on Canny using Scharr, we can see a small improvement regarding the precision (e.g. from 0.282 with Dilated 5x5 to 0.399 with variant 1 in level L1) but we can see a decrease on the recall side with the same amount for the same experiments. This implies a similar overall F1 score and results that provide the equivalence that we expected.

VI. Conclusions and future work

From the experiments and evaluation done in this paper, we can conclude that in most cases, using dilated filters produces similar results as processing at lower scale level and expanding it. Of course, there are benefits when extracting features in lower levels, but we should take in consideration the additional computational steps that appear when we desire to bring edge map back to the initial level.

The results we obtain are encouraging because with a simple dilating operation, we can avoid doing multiple processing steps for eliminating undesired noise from the image. Added with the runtime benefit we show in [Bogdan et al., 2020], we can say that a system using dilated filters has a notable resource gain. The effects of the results are represented by an improvement of the entire system in which an edge detection algorithm is used.

As a future work, one can consider a more complex expansion of the lower pyramid levels edge maps, by using Hierarchical Chamfer Matching as in [Borgefors, 1988], [You et al., 1995], [Zhang et al., 2009], to obtain the same equivalence results for dilated filters.

References

[Adelson et al., 1984] Adelson, E. H., Anderson, C. H., Bergen, J. R., Burt, P. J., and Ogden, J. M. (1984). Pyramid methods in image processing. *RCA engineer*, 29(6):33–41.

[Arbelaez et al., 2011] Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2011). Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916.

[Bogdan et al., 2020] Bogdan, V., Bonchis, C., and Orhei, C. (May 2020). Custom dilated edge detection filters. In *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, page to appear. WSCG.

[Borgefors, 1988] Borgefors, G. (1988). Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on pattern analysis and machine intelligence*, 10(6):849–865.

[Canny, 1986] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698.

[Guizhen et al., 2007] Guizhen, M., Zongliang, F., and Zongzhong, Y. (2007). Performance analysis and comparison of susan edge detector [j]. *Modern Electronics Technique*, 8:189–191.

[Guo and Hall, 1989] Guo, Z. and Hall, R. W. (1989). Parallel thinning with two-subiteration algorithms. *Communications of the ACM*, 32(3):359–373.

[Kawalec-Latała, 2014] Kawalec-Latała, E. (2014). Edge detection on images of pseudoimpedance section supported by context and adaptive transformation model images. *Studia Geotechnica et Mechanica*, 36(1):29–36.

[Kirsch, 1971] Kirsch, R. A. (1971). Computer determination of the constituent structure of biological images. *Computers and biomedical research*, 4(3):315–328.

[Kroon, 2009] Kroon, D. (2009). Numerical optimization of kernel based image derivatives. *Short Paper University Twente*.

[Lai et al., 2017] Lai, W.-S., Huang, J.-B., Ahuja, N., and Yang, M.-H. (2017). Deep laplacian pyramid networks for fast and accurate super-resolution.

[Liu et al., 2019] Liu, X., Deng, Z., and Yang, Y. (2019). Recent progress in semantic image segmentation. *Artificial Intelligence Review*, 52(2):1089–1106.

[Prewitt, 1970] Prewitt, J. M. (1970). Object enhancement and extraction. *Picture processing and Psychopictorics*, 10(1):15–19.

[Prieto and Allen, 2003] Prieto, M. and Allen, A. (2003). A similarity metric for edge images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25:1265– 1273.

[Sasaki, 2007] Sasaki, Y. (2007). The truth of the f-measure. Technical report, School of Computer Science, University of Manchester.

[Scharr, 2000] Scharr, H. (2000). *Optimal operators in digital image processing*. PhD thesis, Ruprecht-Karls-Universitat Heidelberg, Germany.

[Sobel and Feldman, 1973] Sobel, I. and Feldman, G. (1973). *Pattern Classification and Scene Analysis*, pages 271–272.

[Wang et al., 2018] Wang, Y., Zhao, X., Li, Y., and Huang, K. (2018). Deep crisp boundaries: From boundaries to higher-level tasks. *IEEE Transactions on Image Processing*, 28(3):1285–1298.

[Woods, 2011] Woods, J. W. (2011). *Multidimensional Signal, Image, and Video Processing and Coding, Second Edition*. Academic Press, Inc., Orlando, FL, USA, 2nd edition.

[Xu et al., 2011] Xu, Q., Chakrabarti, C., and Karam, L. J. (2011). A distributed canny edge detector and its implementation on fpga. In *2011 Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE)*, pages 500–505.

[You et al., 1995] You, J., Zhu, W., Pissaloux, E., and Cohen, H. (1995). Hierarchical image matching: A chamfer matching algorithm using interesting points. In *Proceedings of Third Australian and New Zealand Conference on Intelligent Information Systems. ANZIIS-95*, pages 70–75. IEEE.

[Zhang et al., 2009] Zhang, Q., Xu, P., Li, W., Wu, Z., and Zhou, M. (2009). Efficient edge matching using improved hierarchical chamfer matching. In *2009 IEEE International Symposium on Circuits and Systems*, pages 1645–1648. IEEE.