Non-Parametric Image Transforms for Sparse Disparity Maps

Dexmont Peña
Dublin City University
Dublin 9, Dublin, Ireland
dpena@computing.dcu.ie

Alistair Sutherland
Dublin City University
Dublin 9, Dublin, Ireland
alistair.sutherland@dcu.ie

Abstract

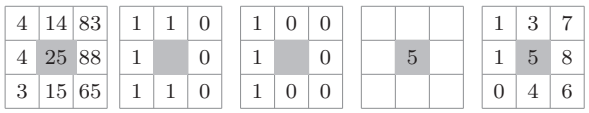
In this paper two image transforms are proposed for the calculation of sparse disparity maps. We present a new variation of the Census Transform, which we call the Thesholded Census Transform. This allows the calculation of the pixels around the edges without a separate edge-detection stage. Then we propose a new application of the Complete Rank Transform (which has so far only been used to calculate optical flow) to solve the Stereo-Matching problem. The utilization of both image transforms represents an improvement in error rates and computational cost against the Census Transform, which is the state of the art image transform used for Stereo-Matching.

1 Introduction

The information from the environment surrounding a mobile robot or autonomous vehicle is crucial for the navigation task. Sensors like laser scanners and time-of-flight cameras provide 3D information in real-time but they are expensive. In addition, this kind of sensor is *active*, meaning they could interfere with other pieces of equipment. An alternative approach to obtain 3D information relies on the use of computer vision to calculate depth maps. The increasing availability of low-cost high-resolution cameras means that this approach is now affordable. In addition, as they are passive sensors, they do not interfere with existing systems.

In stereo vision, the disparity map contains the difference in the location of the objects in each view. When a disparity value is obtained for every pixel in the image it is called a dense disparity map. Approaches based on dense disparity maps are useful for object identification but they are computationally intensive. In contrast, if the disparity is calculated for only a few pixels in the image it is called a sparse disparity map. As fewer pixels are processed in a sparse disparity map, the computational load can be reduced.

Among different approaches, the image transform-based approaches have been shown to be successful for the calculation of dense disparity maps [4, 7, 8]. In this paper a new image transform is proposed for the calculation of sparse disparity maps. In addition, an image transform that has been used up to now only for the calculation of Optical Flow, is proposed for the calculation of disparity maps. The first image transform is the Thresholded Census Transform. This new image transform produces a value only for the pixels, which are located around edges. However it does not need an explicit edge detection. The second image transform is the Complete Rank Transform, which was introduced for the calculation of optical flow in [3]. In this work it is used for matching pixels in the stereo pair.



(a) Value (b) Census (c) TCT (d) Rank (e) CRT

Figure 1: Non parametric and illumination invariant image transforms. a) Intensity values for a window of size 3x3. b)$\xi(p, p')$ for the CT, CT = 11010110. c) $\xi(p, p')$ for the TCT, $\varepsilon = 15$, TCT = 10010100. d) Rank for center pixel, RT = 5. e) Rank for each pixel, CRT = 1,3,7,1,5,8,0,4,6.

This paper is organized as follows: in the next section the work on the Census Transform and Complete Rank Transform for the calculation of disparity maps and optical flow is summarized. In Section 3.1 the Thresholded Census Transform is presented. In Section 4 the experimental results are presented along with the implementation details; and in Section 5 the conclusions are presented.

2 Related Work

In this section a review of the Census Transform and the Complete Rank Transform is presented.

2.1 The Census Transform

The Census Transform (CT), introduced by Zabih and Woodfill [13], computes a bit-string in a square window. The concept starts by defining the function ξ to represent the relationship between the intensity I of the pixels p and p'

$$\xi(p, p') = \begin{cases} 1, & \text{if } I(p') < I(p) \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Then the Census Transform $C(p)$ is defined by creating a vector containing the result of the comparison $\xi(p, p')$ for each pixel p' in the window $N(p)$ centred on p

$$C(p) = \bigotimes_{p' \in N(p)} \xi(p, p') \quad (2)$$

where $\bigotimes$ represents the concatenation operator and the window size is w x w. This transformation has proven to be fast when implemented on embedded systems [7] and requires only a small amount of memory for storing each value. An example of the Census Transform can be seen in Fig. 1b.

Because some of the pixels surrounding the pixel p are similar to it, Stein [12] introduced a similarity threshold ε. This changes the function ξ to:

$$\xi(p,p') = \begin{cases} 0, & \text{if } I(p) - I(p') > \varepsilon \\ 1, & \text{if } |I(p) - I(p')| \leq \varepsilon \\ 2, & \text{if } I(p') - I(p) > \varepsilon \end{cases} \quad (3)$$

Stein used this transform which is called Ternary Census Transform for calculating the Optical Flow. Although more discriminative, this representation loses the small memory footprint of a binary representation by using ternary values. In this paper the concept of the similarity measure is incorporated in the Census Transform whilst keeping a binary representation.

2.2 The Complete Rank Transform

Another transform (presented by Zabih and Woodfill) is the Rank Transform (RT) [13]. The RT is calculated by counting the number of pixels, whose intensity is smaller than the centre pixel p, in a window $N(p)$. This is shown in Fig 1d.

In order to incorporate more information from the window $N(p)$, Demetz, Hafner and Weikert [3] introduced the Complete Rank Transform (CRT). They improve the discriminative power of the RT by calculating the Rank for every item in the support window. This is illustrated in Fig 1e. In [3] the CRT is used to calculate the Optical Flow embedded in a Variational Framework and obtains good results. In this paper the CRT is used for the first time to solve the stereo matching problem.

3 Sparse Disparity Maps

The use of sparse disparity maps reduces the computational load, as the disparity is calculated only for a few pixels in the image. The careful selection of the pixels, for which the disparity is calculated, allows the identification of any possible obstacle in a navigation task. The edge pixels are a good candidate as they are present even in areas with a flat texture.

In recent years the Census Transform has been used for the calculation of Optical Flow [12, 9, 6] and dense disparity maps [7, 10]. This transform has shown its discriminative power is enough to allow the matching of the pixels in the image pair. Although some approaches are claimed to work in real-time, this has been achieved using low resolution images and specialized hardware [7, 10]. It is reasonable to expect that, if the image resolution is increased, the real-time performance would be lost.

In order to reduce the computational load by the use of sparse disparity maps, in this paper a variant of the Census Transform is proposed, which is able to obtain disparity information only from the pixels around the edges. This variant of the Census Transform is introduced in Section 3.1. In addition, the Complete Rank Transform is used for the first time to solve the stereo matching problem in Section 4.2.2.

3.1 The Thresholded Census Transform

Hafner, Demetz and Weickert [6] demonstrated that the good performance of the CT is obtained by extracting information from the areas with high gradient in the image. As is well known, the edges in the image are created by changes in the intensity function. Following this fact, a modification to the CT is proposed.

Table 1: Edge pixels identified by the TCT on the KITTI and Middlebury images using a window size of 16 and similarity $\varepsilon = 6$.

Test image	Canny (%)	EDPF (%)
Average on KITTI	0.82	0.91
Tsukuba	0.83	0.90
Venus	0.85	0.91
Cones	0.87	0.94
Teddy	0.84	0.93
Aloe	0.90	0.94
Flowepots	0.78	0.81

The Thresholded Census Transform (TCT) was inspired by the idea of the Ternary Census Transform [12]. But instead of creating a third value for the similar pixels, the TCT sets their value to 0 as in the Fig. 1c. This changes $\xi(p,p')$ to:

$$\xi(p,p') = \begin{cases} 1, & \text{if } I(p') < I(p) - \varepsilon \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Where ε is the similarity threshold. The use of this similarity threshold allows the incorporation of information from the pixels around the edges while keeping a binary representation. In experiments it was found that the pixels with a non-zero string are located at the brighter side of the edges or in in highly textured regions. The edge detection capability of the TCT is tested in Section 4.1.

4 Experimental Results

In this section the results of the experiments are shown. First the capability of the TCT to identify the pixels around the edges is tested and the results are shown in Section 4.1. Then the TCT and CRT are used to calculate disparity maps and the results are presented in Section 4.2.

4.1 Integrated Edge Detection on the TCT

The edge detection is tested by comparing the pixels, for which a non-zero TCT string is found, against the pixels marked as edges by the Canny [2] and EDFP [1] edge detectors. After calculating the TCT for each image, a mask is created for the pixels which obtained a non-zero string.

The parameters used in the implementation are set to $\varepsilon = 8, w = 9$ as these produced the best performance on the KITTI Benchmark Suite [5] as it will be shown in Section 4.2.1. For the Canny edge detector the parameters 30 and 60 for the lower and higher thresholds are set respectively as in [1]. For EDPF, the implementation available by Akinlar and Topal [1] is taken using a sigma $\sigma = 1$.

Figure 2 shows the average number of edge pixels detected in the images from the KITTI Benchmark Suite. Figure 3 shows that the edges (white pixels) detected by Canny and EDPF are located over the mask produced by the TCT (gray pixels). Table 1 shows the percentage of edge pixels detected on the KITTI and Middlebury [11] datasets.

It is important to note that most of the non-detected pixels are located at one pixel distance from the obtained mask. This indicates the capability of the TCT

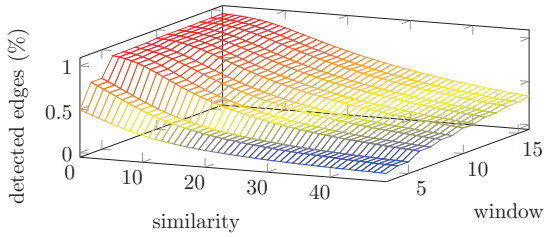


Figure 2: Pixels detected on edges for different parameters of the TCT. The edges were detected by the Canny (x) and EDPF (o) edge detectors.

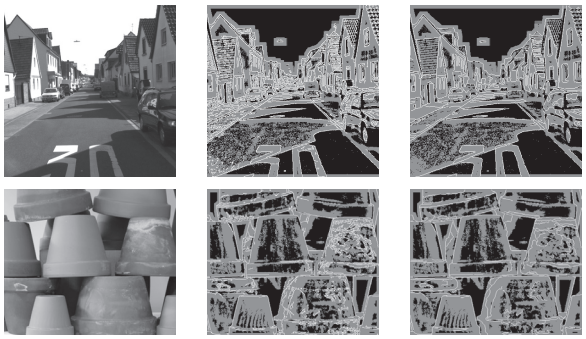


Figure 3: Integrated edge detection on the TCT. From left to right, input image, edges detected by Canny and EDPF respectively. White pixels indicate an edge, gray pixels are the non-zero TCT strings. The shown images are from the KITTI Benchmark Suite and Middlebury dataset respectively from top to bottom.

to capture the edges in the image. Figure 3 shows examples of the detected edges.

The experiments on edge-detection show that for window sizes greater than $w = 8$ there is no significant change in the percentage of detected edge pixels for low similarity values.

4.2 Disparity Calculation

In this section the TCT and CRT are used for calculating the disparity maps on the KITTI Benchmark Suite. This dataset provides semi-dense ground truth for real-world images. All of the experiments were run on a Laptop with a processor Intel Core i7 2675QM at 2.2 GHz and 8GB of RAM using only one thread.

4.2.1 Thesholded Census Transform

The TCT was used for the disparity calculation in the same way the Census Transform is used in [7]. A sparse window size of 9x9 pixels is used along with an aggregation window of 5x5 pixels. The Hamming distance is used for calculating the difference in the values of the transform for a given pixel. As the similarity threshold ε allows the identification of the edges, the texture map required in [7] is not used. In addition, the disparity values are calculated only for those pixels, for which the transform was different from 0.

Different values of the parameters of the TCT were tested on the KITTI Benchmark Suite, in order to find the value that produces the best performance. Figure 4 shows the behaviour of the average disparity error

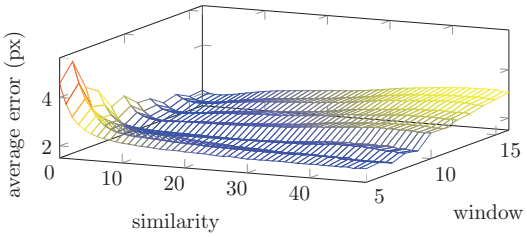


Figure 4: Disparity error vs. similarity ε for the TCT. The minimum value is found when $w = 9$ and $\varepsilon = 8$

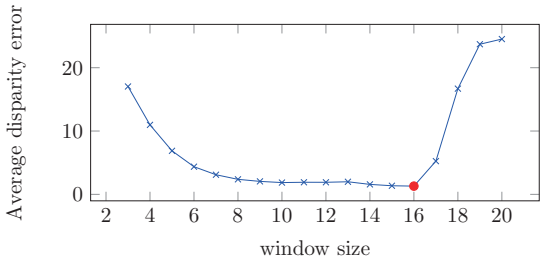


Figure 5: Average disparity error vs. window size for the CRT. The minimum value is found using a window size of 16.

on the KITTI Benchmark Suite at different values of ε and w. It was found that the combination $w = 9, \varepsilon = 8$ minimizes the error. A sample image of the disparity maps obtained by using the TCT is shown in Figure 6.

4.2.2 Complete Rank Transform

The implementation of the CRT for calculating the disparity map is based on the one used in [7]. But the sparse window is not used, i.e., all of the pixels in the window are required for the calculation of the transformation. In addition, as the obtained values in the transform are integer, the Sum of Absolute Differences (SAD) is used as cost function. No aggregation window was used for the cost calculation, as it was found that it increased the computational load without increasing the quality of the disparity maps.

Different window sizes were tested on the KITTI Benchmark Suite. Figure 5 shows the effect of the window size on the accuracy of the obtained disparity maps. A window size of 16 pixels was found to produce the most accurate disparity maps. A sample disparity map obtained by the CRT is shown in Figure 6.

In order to compare the performance of the proposed methods, the TCT and CRT are compared against the Census Transform as presented in [7]. The obtained results are shown in Table 2. From this table it can be seen that the CRT produced denser disparity maps although the running time is higher than the binary transforms. Although the TCT obtained the sparser disparity maps, these contain information from the pixels around the edges, allowing the visual identification of the obstacles in a navigation task.

5 Conclusions and Future Work

Two image transforms invariant to changes in illumination are introduced for the calculation of disparity

Table 2: Comparison of the average disparity error for the CT, TCT and CRT.

	Error threshold					Density	Avg. Error	Running time (s)
	1	2	3	4	5			
CT [7]	0.131755	0.080045	0.068259	0.062245	0.058101	0.35045	2.127764	2.34
TCT	0.104788	0.063885	0.056033	0.051821	0.04877	0.140475	1.845432	1.73
CRT	0.072721	0.047265	0.041529	0.038324	0.035663	0.066979	1.300896	6.34

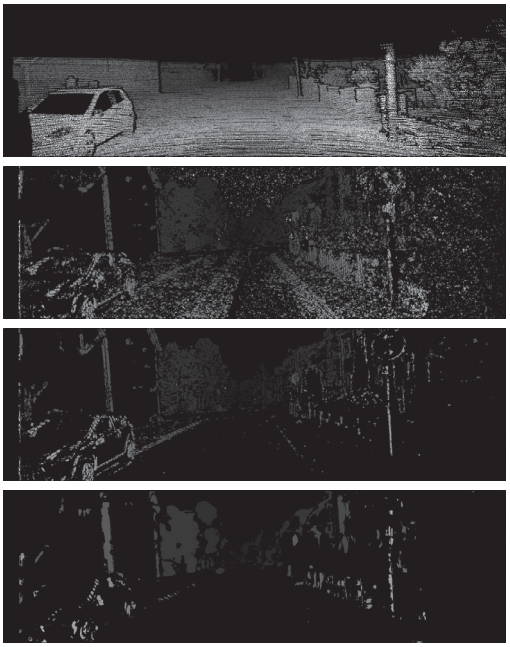


Figure 6: Disparity maps obtained by the different image transforms. From top to bottom the transforms are: Ground Truth, Census Transform, Thresholded Census Transform and Complete Rank Transform respectively.

maps. The TCT was shown to successfully extract information from the pixels around the edges without a separate edge-detection stage. The CRT was shown to increase the density of the obtained disparity maps at the cost of longer computation. Both image transforms present an improvement over the original Census Transform in the presented average error and the computational cost.

Further research will include the use of an adaptive similarity measure for the TCT and the incorporation of the new image transforms into an edge-aware algorithm for disparity calculation.

Acknowledgments

This research has been funded by the Irish Research Council under the EMBARK initiative, application No RS/2012/489.

References

[1] Cuneyt Akinlar and Cihan Topal. Edpf: a Real-Time Parameter-Free Edge Segment Detector With a False Detection Control. *International Journal of Pattern Recognition and Artificial Intelligence*, 26(01):1255002, February 2012.

[2] John Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, November 1986.

[3] Oliver Demetz, David Hafner, and J Weickert. The complete rank transform: A tool for accurate and morphologically invariant matching of structures. In *24th British Machine Vision Conference*, 2013.

[4] Wade S Fife and James K Archibald. Improved Census Transforms for Resource-Optimized Stereo Vision. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(1):60–73, January 2013.

[5] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, June 2012.

[6] David Hafner, Oliver Demetz, and Joachim Weickert. Why Is the Census Transform Good for Robust Optic Flow Computation? In *Scale Space and Variational Methods in Computer Vision*, volume 7893 of *Lecture Notes in Computer Science*, pages 210–221, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[7] Martin Humenberger, Christian Zinner, Michael Weber, Wilfried Kubinger, and Markus Vincze. A fast stereo matching algorithm suitable for embedded real-time systems. *Computer Vision and Image Understanding*, 114(11):1180–1202, November 2010.

[8] Xing Mei, Xun Sun, Mingcai Zhou, Shaohui Jiao, and Haitao Wang. On building an accurate stereo matching system on graphics hardware. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 467–474. IEEE, November 2011.

[9] Thomas Müller, Clemens Rabe, Jens Rannacher, Uwe Franke, and Rudolf Mester. Illumination-Robust Dense Optical Flow Using Census Signatures. In *Pattern Recognition*, volume 6835 of *Lecture Notes in Computer Science*, pages 236–245, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[10] Stefania Perri, Pasquale Corsonello, and Giuseppe Cocorullo. Adaptive Census Transform: A novel hardware-oriented stereovision algorithm. *Computer Vision and Image Understanding*, 117(1):29–41, January 2013.

[11] Richard Scharstein, Daniel and Szeliski, D Scharstein, R Szeliski, and R Zabih. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, 47(1):7–42, 2002.

[12] Fridtjof Stein. Efficient Computation of Optical Flow Using the Census Transform. In *Pattern Recognition*, volume 3175 of *Lecture Notes in Computer Science*, pages 79–86, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[13] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In Jan-Olof Eklundh, editor, *Computer Vision ECCV '94*, volume 801 of *Lecture Notes in Computer Science*, pages 151–158, Berlin/Heidelberg, 1994. Springer-Verlag.