

Evaluate the Gray Code in Distributed Fields for Tracking

Xianyun Wu^{1,a}, Kai Liu^{2,b}, Jinyan Cao^{3,c}, Yunsong Li^{4,d} and Li Wang^{5,e}

^{1,2,3,4}State Key Lab. of Integrated Service Networks, Xidian University,
Xi'an, China

^{1,2}School of Computer Science and Technology, Xidian University, Xi'an, China

^axywu@mail.xidian.edu.cn, ^bkailiu@mail.xidian.edu.cn,

^chappyjinyancao@163.com, ^dysliu@mail.xidian.edu.cn,

^ewangli.xidian@gmail.com

Abstract

Distribution Fields (DFs) for tracking achieved a better performance than traditional algorithms due to its special representation that allows smoothing the objective function without destroying information about pixel values. DFs descriptor can satisfy both the specificity and smooth landscape requirements of a good tracking algorithm. In this paper, we evaluate the Gray code in original DFs algorithm by replacing the pixel values using Gray code instead of original binary code. Experimental results show that the Gray code can improve the tracking efficiency in certain way.

Keywords: Distribution Fields (DFs) tracking Gray code

1. Introduction

Traditional tracking algorithms usually smooth the objective function to blur the image to guarantee the gradient descent of the alignment function reach the global optimum. However, blurring the image destroys image information. Recently, Laura and Erik proposed a new representation in target tracking called Distributed Fields(DFs). DFs descriptor can satisfy both the specificity and smooth landscape requirements of a good tracking algorithm without higher order statistics or temporal information [9], and using feature selection [10-12]. The kernel-based framework of DFs includes spatial information in it which resolves ambiguity and overcomes the under-sensitivity to spatial structure. It has a wider basin of attraction around a target's location and tracker built from this descriptor also outperforms other state-of-art trackers.

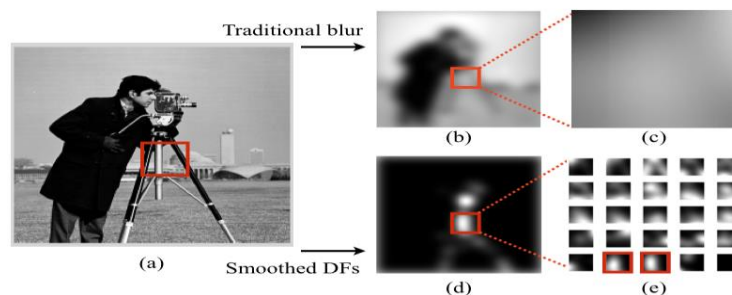


Figure 1. Information Preserved Using Smoothing on a DF. (a) An Original Cameraman Image. (b) Image Smoothed with Traditional Blur. (c) Patch of Image (b) Where There Central Bar Ysed to be. (d) Layer of the DF Corresponding to the Intensity Value of the Bar. (e) Collection of Patches of the DF under the Location of the Central Bar. When an Image is Blurred, the New Pixel Values are a Combination of the Neighboring Pixels Around them, and All the Information is Collapsed into a Single Number

2. Description of Distribution Fields

A distribution field (DF) is simply an array of probability distributions, one for each location in a “field”. The probability distribution at each location defines the probability of each feature value at the location.

For example, if the feature space is gray-scale intensity, then at each pixel there is a probability distribution over the values 0-255.

2.1. Representation

A DF is represented as a matrix d with $(2+N)$ dimensions, where the first two dimensions are the width and height of the image, and the other N dimensions index the feature space that we choose. For example, if the feature space is intensity, then an image of size $m \times n$ yields a 3D DF of size $m \times n \times b$, where b is the number of intensity feature values, or bins. For a higher dimensional feature space, such as two-dimensional gradient measurements or normalized RG color distributions, we can build a 2D distribution at each pixel location, yielding a DF of four dimensions. It can be describes as following:

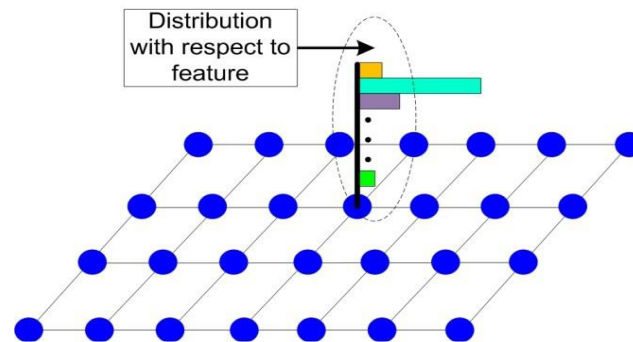


Figure 2. The Feature Space is Gray-Scale Intensity, then at Each Pixel there is a Probability Distribution Over the Values 0-255

Therefore the distribution field of an image patch yields a high-order tensor. Some multiple channel filter techniques (*e.g.*, Gabor and PPBTF) can be viewed as distribution fields if they are normalized (PS: probability distribution: non-negative values, the sum of all values is equal to one).

2.2. Construction

Exploding an image into a DF results in a Kronecker delta function at each pixel location. In particular, exploding an image I into d with as many bins as features values is defined by

$$d(i, j, k) = \begin{cases} 1 & \text{if } I(i, j) = k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where i and j index the row and column of the image, and k indexes the possible values of the pixel. We call the collection of bins at a fixed depth k a layer. This produces a probability distribution at each pixel since the sum of the components of each column is 1. The left side of Figure 3, shows the results of computing this DF for the well-known “cameraman” image. At this point, the DF representation contains exactly the same information as the original representation, albeit in a larger representation.

The bilateral filter [7] introduced a way of smoothing an image such that both proximity in space and feature value are taken into account to preserve image detail,

which is similar to blurring using DFs. The blurring of distribution fields has the same benefits as blurring images [8], but few of the drawbacks. No information has been lost about the value of pixels in the original image. We now show how to “spread” the information in the image without destroying the brightness values as occurs with traditional blurring. The right side of Figure 3, shows a smoothed version of the DF on the left. The 3D DF has simply been convolved with a 2D Gaussian filter which spreads out in the x and y dimensions, but not in the feature dimension. That is, each layer k of the smoothed DF df_s is computed as

$$df_s = df * h(x, y) \quad (2)$$

where $h(x, y)$ is a 2D Gaussian kernel of standard deviation σ_s , and “ * ” is the convolution operator.

Prior to convolution, we could interpret any value of 1 in layer L of a DF to mean “there is a pixel of value L at this location in the original image.” After convolution, the semantics of the smoothed DF is, for any non-zero value in a layer L, “there is a pixel of value L somewhere near this location in the original image.” Thus, the convolution has introduced positional uncertainty into the representation. A

critical point is that no information has been lost about the value of pixels in the original image, only about their position. This is because there has been no mixing of pixel values during the convolution process.

The feature convolution also has introduced some uncertainty. This allows the model to explain small changes due to subpixel motion, shadows, and changes in brightness. In a grayscale image, this smoothing is a 1D Gaussian filter over the third dimension. Each of the columns of df_s can be smoothed to produce df_{ss} as

$$df_{ss} = df_s * h(z) \quad (3)$$

where $h(z)$ is a 1D Gaussian kernel of standard deviation σ_f .

In summary, exploding an image into a DF and smoothing it can be viewed as introducing uncertainty about the object appearance. A DF is then a compact representation of the image itself and a set of its “neighboring” images. These images are the result of transforming the original image with small changes in appearance and in location. These are weighted according to the simple assumption that the most likely event is that the image will stay the same, and larger changes are less likely.

2.3. Comparison

The comparison between DFs that different images yield can be done with any distance function. In this paper we use the L1 distance between the two arrays d_1 and d_2 as:

$$L_1(d_1, d_2) = \sum_{i,j,k} |d_1(i, j, k) - d_2(i, j, k)| \quad (4)$$

Searching for the target in a new frame consists of building a new DF by also exploding and smoothing the new frame, and following the direction where the gradient of the L_1 descends.

2.4. Combination

Combining the information of several DFs can also be useful. In tracking we combine the DF of initial model and the DFs of new observations using a component-wise convex combination of them, which also yields a DF:

$$d_{t+1}(i, j, k) = \lambda d_t(i, j, k) + (1 - \lambda) d_{t-1}(i, j, k) \quad (5)$$

By combining DFs of different instances of the same object we build a non-parametric data-driven model of the distribution at each pixel.

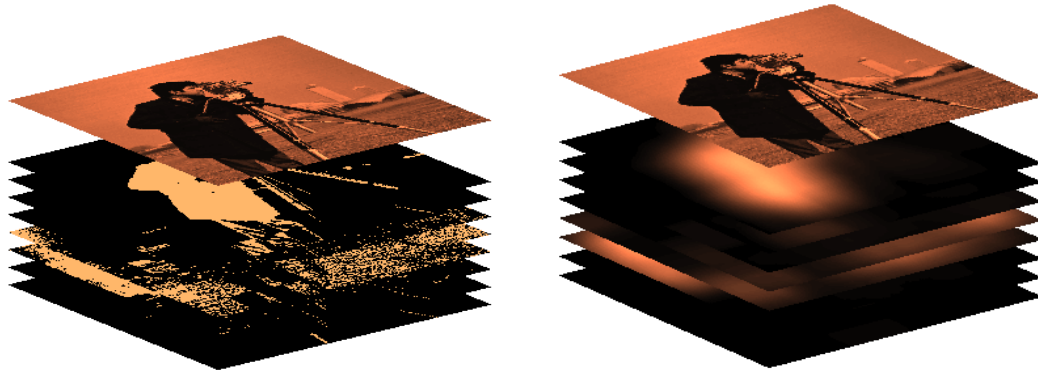


Figure 3. Left. Exploding an Image into a Distribution Field. (The Original Image is Shown Superimposed on the DF for Clarity.) The Number of Brightness Levels (or Layers) has been Quantized to 8 with Respect to Feature Values. Right. Smoothing Distribution Fields

3. Gray Code in Distributed Fields for Tracking

Before tracking, we will convert the binary code into gray code. A model of the target is created by exploding the image that contains the target into a DF and smoothing it. Searching for the target in a new frame consists of building a new DF by also exploding and smoothing the new frame, and following the direction where the gradient of the L1 difference between the DF of the model and the underlying part of the bigger DF representing the new frame descends. Once a local minimum is reached, the model of the target is updated, using a linear combination of the model and the new observation, as in Equation 4. For better performance, we use a hierarchical approach. Instead of using a single DF to represent the target, we use a small set of DFs, where each of them is built using an increasing value of the parameter σ_s , which regulates the amount of spatial blur. These DFs contain information at different frequencies. At each frame, we use a coarse-to-fine strategy. The most smoothed DF is used to start the search, until it reaches a local minimum. This position is the start for the search in the second DF.

Parameter b corresponding to the number of bins was chosen, for speed, as the smallest power of two that does not hurt the performance of the videos. This is $b = 16$.

$\sigma_f = 10$ is also the case for $\lambda = 0.95$, whose method for choosing the value of the two parameters is using leave-one-out cross validation.

Algorithm 1 Tracking with distribution fields

Input: $V =$ video sequence.

I = patch containing target in frame 1.

$\sigma_s =$ set of spatial smoothing parameters.

$\sigma_f =$ brightness smoothing parameter .

b = number of brightness bins ($b = 16$).

$\lambda =$ mixing parameter ($\lambda = 0.95$).

Output: $(x, y)_f$ {Positions of target at each video frame f in V }

1. Initialize convert binary code into gray code

2: Initialize $d_{model}^i = explode(I) * h_{s(i)} * h_f, i \in 1, \dots, |\sigma_s|$

3: Initialize target location (x, y) to center of patch **I**.

```

4: for f = 2 ->|V| do
  5: for i = 1 ->|σs| do
    6: dfi = explode(f) * hs(i) * hf
    7: (x', y') = arg min(x,y) L1(dfi(x, y), dmodeli)
    8: (x, y) = (x', y')
  9: end for
  10: dmodel = λdmodel + (1 - λ)df(x, y)
11: end for

```

4. Gray Code in Distribution Fields for Tracking

In this paper, we integrate the Gray code into DFs descriptor. Experimental results show that the Gray code can improve the tracking efficiency for a large part of standard tracking data sets.

Table 1. Original Code and Gray Code by Bit Width

Value	Original Code	Gray Code
0	0000 0000	0000 0000
1	0000 0001	0000 0001
2	0000 0010	0000 0011
3	0000 0011	0000 0010
4	0000 0100	0000 0110
5	0000 0101	0000 0111
6	0000 0110	0000 0101
7	0000 0111	0000 0100
...

4.1. Proposed Descriptor

The Gray code [2] is a binary numeral system where two successive values differ in only one bit (binary digit) as shown in Table 1. Gray code has been widely used in many fields. For example, in hardware design it can be used to eliminate the error occurs between logic states transition. In this paper, we use the Gray code instead of the original code to construct the DF of the target area and the image. The construction of DF layers still use previous DFs method [1].

To reduce the complexity of Gray code generation, we use look-up table for Gray code indexing. The Gray code of original code is stored in fixed look-up table. The index of Gray code can directly located by the original code value which only increase the memory usage without increase the computation. After construct the Gray code based DF layers, we perform 2D and 1D Gaussian filtering in spatial and brightness respectively. We use the same tracking algorithm and the same parameters as depicted in paper [1].

4.2. Experimental Results

The proposed descriptor is implemented in DFs reference software [6]. All the tracking data sets are from standard library. We compare the performance of our algorithm to other four algorithms: original DFs [1], MIL [3], PROST [4] and MKT [5]. Their experimental results are simply taken from paper [1]. We use two different metrics for the analysis of the tracking results: the percentage of frames correctly tracked Table 2, and the mean distance to the ground truth Table 3. It can be seen from Table 2, our proposed algorithm outperform the original DFs algorithm for all the test series except the video "sylvester".

The performance of video "girl" is greatly improved which cause our algorithm even ourperform the PROST.

Table 2. Percentage of Correctly Tracked Frames

Video	Gray DFs	DFs[1]	PROST	MIL	MKT
david	100.00	100.00	80	58.91	30.43
sylvester	61.71	66.79	74	73.88	20.90
girl	97.03	73.00	89	55.20	7.00
faceocc	100.00	100.00	100	77.28	6.21
coke11	76.27	75.86	—	17.93	27.59
dollar	100.00	100.00	—	90.76	15.38
cliffbar	87.88	87.69	—	72.31	—
twinnings	69.15	69.15			

Table 3. Mean Distance to the Ground Truth

Video	Gray DFs	DFs[1]	PROST	MIL	MKT
david	10.24	9.97	15.30	23.45	98.62
sylvester	28.00	15.92	10.60	10.62	49.24
girl	20.99	21.57	19.00	32.76	105.05
faceocc	4.14	5	7.00	27.28	102.47
coke11	6.63	7.19	—	20.85	20.33
dollar	5.23	5.26	—	15.15	81.26
cliffbar	7.25	7.77	—	12.23	—
twinnings	12.75	13.20			

4.3. Result Analysis

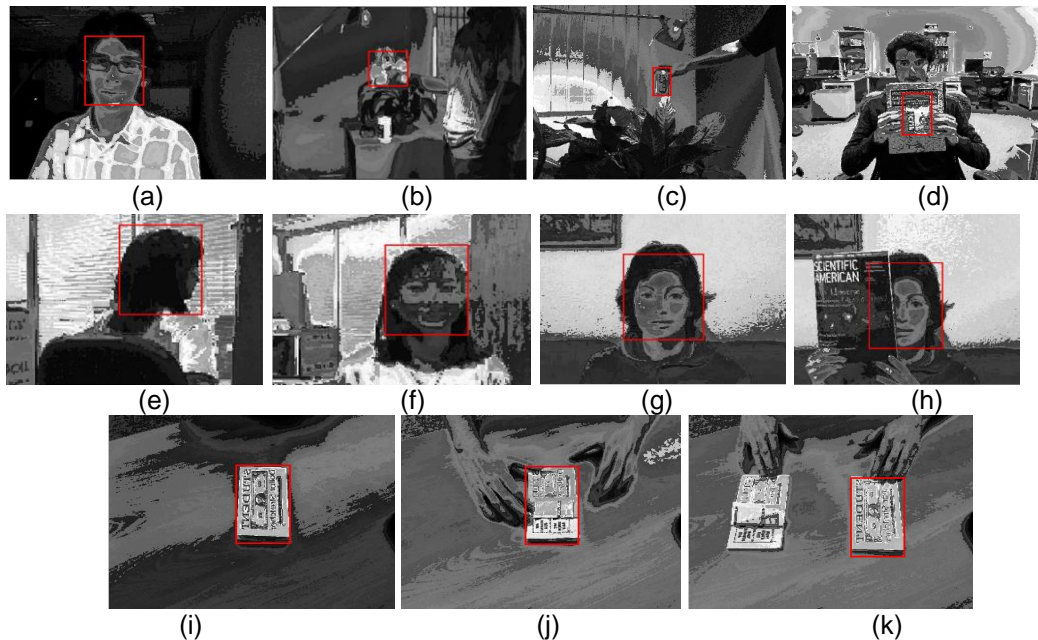


Figure 4. Sample Frames. Overcome Limited Occlusions (g,h), Moderate Changes in Illumination and the Light (a,b,c,d). Drift Darstic(e,f), and the Changes in Appearance(i,j,k)

5. Conclusion

In this paper, an improved Gray code based DFs descriptor is proposed. It achieved a better tracking performance for most tracking data sets. The experimental results show that the proposed representation can be a good complementation of current DFs algorithm. This paper also remind us a good assumption that the descriptor which can keep more information in each layer will achieve a better performance.

Acknowledgments

This work was supported in part by the China Postdoctoral Science Foundation(2013M540735), National Natural Science Foundation (grant 61222101, 61072065, 61007011, and 61301291), Fundamental Research Funds for the Central Universities, and the "111" project (B08038).

References

- [1] L. S. Lara and E. L. Miller, "Distribution Fields for Tracking", Computer Vision and Pattern Recognition (CVPR), (2012), pp. 1910-1917.
- [2] F. Gray, "Pulse Code Communication", U.S. Patent 2 632 058, (1953) March 17.
- [3] B. Babenko, M. H. Yang and S. Belongie, "Visual tracking with online multiple instance learning", Computer Vision and Pattern Recognition(CVPR), (2009), pp. 983-990.
- [4] J. Santner, C. Leistner and A. Saffari, "PROST: Parallel robust online simple tracking", Computer Vision and Pattern Recognition(CVPR), (2010), pp. 723-730.
- [5] G. Hager, G. D. Hager, M. Dewan and C. V. Stewart, "Multiple kernel tracking with SSD", Computer Vision and Pattern Recognition(CVPR), (2004), pp. 790-797.
- [6] <http://people.cs.umass.edu/~lsevilla/trackingDF.html>.
- [7] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images", In ICCV, (1998).
- [8] H. Mobahi, C. L. Zitnick and Y. Ma, "Seeing through the blur", In CVPR, (2012).
- [9] K. J. Cannons, J. M. Gryn and R. P. Wildes, "Visual tracking using a pixelwise spatiotemporal oriented energy representation", In EECV, (2010).
- [10] R. T. Collins and Y. Liu, "On-line selection of discriminative tracking features", In ICCV, (2003).
- [11] A. P. Leung and S. Gong, "Mean shift tracking with random sampling", In BMVC, (2006).
- [12] Z. Yin, F. Porikli and R. T. Collins, "Likelihood map fusion for visual object tracking", In IEEE Workshop on Applications of Computer Vision, (2008).

Authors



Xianyun Wu was born in 1985. He is now a lecturer at Xidian University. He received his PhD degree from Xidian University in 2012. His research interests include remote sensing, image compression and processing, high-performance computing, and distributed source coding.



Kai Liu He received the B.S. and M.S. degrees in computer science and the Ph.D. degree in signal processing from Xidian University, Xi'an, China, in 1999, 2002, and 2005, respectively. Currently, he is an Professor of computer science and technology with the Xidian University. Professor Liu has attended China lunar explore project from Chang'er-1,2,and 3 which he has been chief designer for the core part of image compression hardware in the Chang'er-1 satellite. He has received several awards from different organizations. These are Shaanxi science and technology award by

province government in 2009, the best teacher award from EMC corporation in 2009,2010, the excellent cooperation award from Huawei corporation for his work on the GSM/UTMS data compression in 2013, the excellent cooperation teacher award from IBM corporation for his work on IBM-Xidian education project and SSD data processing project in 2014. He has published more than 30 papers on image/data compression. His main research interests are in the area of FPGA/ASIC design for image and text coding.



Jinyan cao received a Bachelor's degree from the Institute of Communication Engineering, Xidian University, Xi'an, China, in July 2013. She is currently working toward a Master's degree in communication and information systems from Xidian University. Her research interests are the areas of image tracking.



Yunsong Li received an M.S. degree in telecommunication and information systems and a Ph.D. degree in signal and information processing from Xidian University, China, in 1999 and 2002, respectively.

He is currently a Professor at the State Key Laboratory of Integrated Service Networks at Xidian University, where he advises M.S. and Ph.D. students. He has authored or coauthored over fifty scientific and technical publications.

Dr Li currently serves as a Committee Member of the Chinese Lunar Exploration Program and has been serving as a co-chair for the SPIE Conference on Satellite Data Compression since 2012. His research interests are in the fields of image and video compression and communications, image processing in remote sensing, and high-performance computing in image processing.



Li Wang received an M.S. degree in telecommunication and information systems from Xidian University in 2012. She is currently working toward a Doctor's degree in telecommunication and information systems from Xidian University. Her research interests are the areas of image tracking and processing, wireless communication.