MULTIPLE HISTOGRAM MATCHING

Dori Shapira, Shai Avidan

Faculty of Engineering
Tel-Aviv University

Yacov Hel-Or

School of Computer Science
The Interdisciplinary Center (IDC), Herzliya

ABSTRACT

Histogram Matching (HM) is a common technique for finding a monotonic map between two histograms. However, HM cannot deal with cases where a single mapping is sought between two *sets* of histograms. This paper presents a novel technique that finds such a mapping in an optimal manner under various histograms distance measures.

1. INTRODUCTION

In many scientific disciplines there is a need to determining a monotonic mapping between two discrete sequences (i.e. a mapping that maintains the internal order of the sequence elements). A classic example is *color calibration* between two images. In this case, one wishes to remap the tones (intensities) of one image to another image, while maintaining the shadow regions darker than the highlights.

When two images are captured from the same view-point but at different times or using different camera parameters, color calibration between the images can be found directly from their color pixels [1]. In this case, one wishes to find a monotonic color map that optimally fits the pixel colors in the source image to their corresponding pixel colors in the target image. As pixel correspondences between the two images are used for calculating the color map, spatial information is exploited. We consider such solutions as *spatial* approaches.

In more challenging cases, the images were acquired under different view-points, thus, pixel correspondence is not directly available and difficult to extract. In such cases, color mapping is commonly calculated from the images' histograms where only statistical information is considered while spatial information is ignored. We consider solutions of this type as *statistical* approaches. Clearly, such a solution is inferior to the spatial approaches and may suffer from inaccuracies.

In this paper we propose to exploit spatial as well as statistical information for color calibration. The main idea is to divide the images into a set of local regions and construct a *local histogram* for each region. Such local histograms are insensitive to small geometric deformations. We construct a *histogram correspondence* such that for each local histogram in one image a corresponding histogram in the other image is determined. Histogram correspondence can be obtained either by calculating a rough geometric transformation between the images or by corresponding image regions using local features such as SIFT or SURF [2, 3]. Given two sets of local histograms, one set for each image, along with a correspondence between the histograms, an optimal monotonic color mapping is calculated to simultaneously satisfy the entire set of histogram pairs. This approach enables, on one hand to handle inaccurate geometric transformation by considering statistical information (histograms), while on the other hand to exploit spatial information as the histograms describes local statistics.

Histogram Matching (HM) [4, 5] is a common approach for finding a monotonic mapping between a pair of histograms. Given two histograms (PDFs) the algorithm finds a color mapping that optimally transforms one histogram towards the other. HM is a simple yet very effective algorithm, however it suffers from a number of shortcomings: First, it is limited to only two histograms and cannot deal with multiple histograms simultaneously. Second, HM approximates the optimal solution with respect to the L1 norm over the cumulative histogram pair [6, 7, 8], but is unable to provide an optimal solution for other metrics. Finally, the HM solution is designed for continuous PDFs and may produce non optimal solutions in the discrete histogram case.

In this paper, we propose a new algorithm that generalizes HM in a number of ways. First, the algorithm can find a single monotonic mapping between *multiple pairs* of histograms such that the mapping will satisfy all pairs simultaneously. Second, the algorithm can work with any distance metric as long as it is additive (such as L_1, L_2, KL, χ^2, histogram intersection, etc.). Finally, the algorithm can work with distance metrics defined over histograms as well as cumulative histograms providing the optimal solution. Since the algorithm generalizes the traditional Histogram Matching we term it *Generalized Histogram Matching* (GHM). Formally, we consider the following scenario:

1. Given two images I_a and I_b we divide the two images into k distinct sub-images. For each sub-image we calculate a local histogram resulting in two sets of histograms $\mathcal{A} = \{\mathbf{a}_i\}_{i=1}^k$ for image I_a and $\mathcal{B} = \{\mathbf{b}_i\}_{i=1}^k$ for image I_b, such that $\mathbf{a}_i$ corresponds to $\mathbf{b}_i$, $i = 1..k$. We do not constrain corresponding histograms to have the same number of bins,

thus the two sets can discretize the continues range of gray-values with different quantizations bins.

2. A distance measure $d(\cdot,\cdot)$ between two histograms (with the same number of bins) is given. We assume d is additive, i.e. for any two histograms, $\mathbf{a}$ and $\mathbf{b}$, $d(\mathbf{a},\mathbf{b}) = \sum_i d(a(i),b(i))$, where $d(a(i),b(i))$ is a bin-to-bin distance[1] Accordingly, for two sets of histograms $\mathcal{A}$ and $\mathcal{B}$, the distance is defined as a sum of distances over all histogram pairs:

$$d(\mathcal{A},\mathcal{B}) = \sum_{\mathbf{a}_i\in\mathcal{A},\mathbf{b}_i\in\mathcal{B}} d(\mathbf{a}_i,\mathbf{b}_i)$$

3. W.l.o.g. we seek the optimal color monotonic mapping $\mathcal{M}$ applied to image I_a so that its resulting sub-image histograms $\mathcal{M}(\mathcal{A})$ will be as close as possible to $\mathcal{B}$. Formally speaking, we are looking for $\mathcal{M}$ minimizing the following distance:

$$E(\mathcal{M}) = d(\mathcal{M}(\mathcal{A}),\mathcal{B}))$$

In the following we present an efficient method for finding the optimal monotonic mapping $\mathcal{M}^*$ under a given distance measure d. The algorithm minimizes the (possibly weighted) sum of distances over all histogram pairs at once using dynamic programming. We do not restrict the distance measure to be l_1 or l_2 norms and we can deal with any additive distance. Additionally, the approach is easily extended to deal with distances defined between two sets of cumulative histograms (CDFs). Such distances are especially interesting as they were shown to be analogous to the Earth Mover Distance (EMD) in the 1D case [9, 8, 7].

2. MAPPING AS ROW TRANSFORMATION

Let the two sets of histograms $\{\mathbf{a}_i\}_{i=1}^k$ and $\{\mathbf{b}_i\}_{i=1}^k$ be represented by two matrices as follows: $A = [\mathbf{a}_1,\mathbf{a}_2,\cdots,\mathbf{a}_k]$ and $B = [\mathbf{b}_1,\mathbf{b}_2,\cdots,\mathbf{b}_k]$. We do not constrain the two sets to have the same number of bins, thus, A is a $n\times k$ matrix and B is $m\times k$. We define an auxiliary matrix $\tilde{A}$ to be of same size as B. We will apply the mapping operator from A to $\tilde{A}$, while aiming to minimize the distance $d(\tilde{A},B)$. Mapping the tone-value of bin i in image I_a to the tone-value of bin j in image I_b implies that the entire row i from A is mapped to row j in $\tilde{A}$. Thus, color transformation applied to I_a can be regarded as row transformations applied to A and resulting in $\tilde{A}$. Observe that multiple source rows are allowed to aggregate into a single destination row (many-to-one mapping), however, splitting a source row into several target rows is not allowed. If the mapping does not map any of the source rows to any particular target row, this target row remains zero.

The above definition of row mapping allows us to introduce the mapping operator $\mathcal{M}$ as a matrix multiplication:

$$\tilde{A} = MA$$

[1]To simplify notations we denote bin distance and histogram distance with the same symbol d.

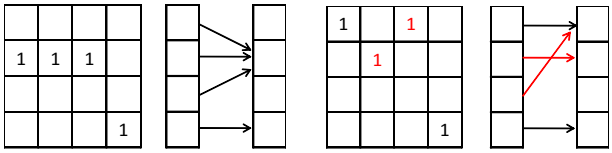

Fig. 1. Left: A monotonic mapping matrix along with the mapping results drawn with arrows. Right: A non monotonic mapping matrix and its arrows. In this example $m = n = 4$, and $k = 1$.

where M is an $m\times n$ matrix, whose entries are all zeros, except for a single element in every column, whose value is 1. We will call this element the column's *indicator*. An indicator existing at $M(i,j)$ will accumulate the entire row j in A into row i in $\tilde{A}$. We denote by $\pi_M(j)$ the entry number in which the indicator appears in column j, namely, $M(i,j) = 1$ gives that $\pi_M(j) = i$. Since mappings are restricted to be monotonic we require that

$$\pi_M(j) \geq \pi_M(j') \quad \text{for } j > j' \tag{1}$$

i.e. if any row j in A is mapped to some row i in $\tilde{A}$, then all rows $j' < j$ must be mapped to rows $i' \leq i$ (see Figure 1). Thus, matrix M can be seen as a matrix of zeros, except for a "seam" of indicators, crossing it from left to right in a non-increasing manner. Denote by $\mathcal{S}$ the set of all monotonic matrices as defined above. Our goal is to find an optimal mapping matrix $M\in\mathcal{S}$ that minimizes the objective function:

$$M^* = \arg\min_{M\in\mathcal{S}} d(MA,B) \tag{2}$$

3. THE GENERALIZED HISTOGRAM MATCHING

As defined above, each column of the mapping matrix M must have exactly one indicator. Due to the monotonicity of M, each row must satisfy the following two conditions: First, given an indicator existing at $M(i,j)$, no indicator can appear neither at the lower-left sub-matrix of $M(i,j)$, nor at its upper-right sub-matrix (see Figure 2-c). The second condition is that if there are several indicators in a certain row, they must be consecutive, otherwise the previous condition is violated.

The above two conditions of the mapping matrix M allows for an efficient calculation of the optimal path of indicators. The outline of the algorithm goes as follows:

1. During the process two auxiliary matrices, T and C, are maintained. The matrices are of the same size as M. We use T as the *trace-back* matrix, helping to keep track of the optimal path of indicators, while C is used as the *cost* matrix such that each cell, $C(i,j)$, represents the best cost achievable by a partial path of indicators starting at the first row and ending at the i^{th} row, given that the right-most indicator in

this path is at column j.

2. A row-by-row scan of matrix C is performed, starting at row $i = 1$, and calculating the cost values for every entry $C(i, j)$. In the first row, $C(1, j)$ is evaluated as follows:

$$C(1, j) = RowCost(1, 1 \cdots j) \quad \text{and} \quad T(1, j) = 0$$

where *RowCost* is defined as:

$$RowCost(i, j \cdots k) = d\left(\sum_{p=j}^{k} A(p, \cdot)\ ,\ B(i, \cdot)\right)$$

That is, $RowCost(i, j \cdots k)$ calculates the additional cost of mapping rows $j \cdots k$ in A into row i in B. In matrix M this is represented as a sequence of consecutive indicators in $M(i, j \ldots k)$ (see Figure 2-b).

3. For the following rows, each entry $C(i, j)$, $i = 2 \ldots n, j = 1 \ldots m$, is calculated by seeking the optimal indicator path admissible by $C(i, j)$. At this point we exploit the additivity of the distance measure and calculate the optimal path using the values in $C(i - 1, \cdot)$. For each $C(i, j)$, the optimal connecting indicator path starting at row $i-1$ and ending at row i is sought (Figure 2-b). This search is performed by scanning all values of $C(i - 1, j')$, $1 \leq j' \leq j$, and adding the cost for completing the path with consecutive indicators in row i:

$$C(i, j) = \min_{j' \leq j} \{C(i - 1, j') + RowCost(i, (j' + 1) \ldots j)\}$$

In order to keep track of the optimal path, the optimal index of j' is stored in the trace-back matrix, at $T(i, j)$. Since evaluating a row in T and C only requires data from the previous row, C and T can be evaluated using a linear top-down and left-to-right scan. Figure 2-b shows a situation where for $C(i, j)$, the optimal connecting indicator path was chosen to initiate from $(i - 1, j')$. Thus, $T(i, j)$ was associated with j'. As a result, if location (i, j) will be included in the final path, a sequence of $j - j'$ indicators will be inserted in the i'th row of M.

4. The process terminates when all costs are filled in C. The optimal cost over all possible paths can be determined from $C(m, n)$, and the optimal path of indicators is constructed by tracing back from $T(m, n)$ up to the first row of T. The transformation matrix M is then populated by filling in indicators according to the constructed path.

Since the distance is assumed to be additive, partially optimizing the distance function (over only some of the rows of M) still leads to a global minimum of the total cost at the end of the procedure, as performed in standard Dynamic Programming schemes. Therefore, this algorithm can work with any additive distance and outputs the optimal transformation matrix.

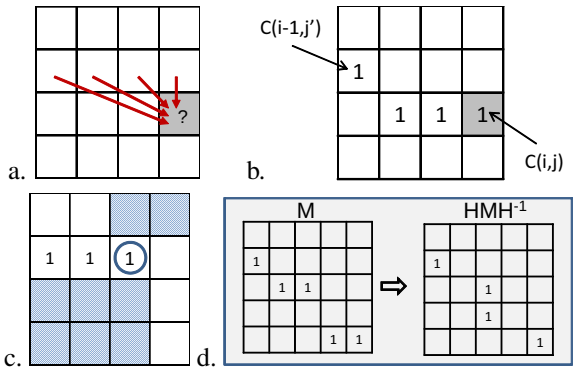

Fig. 2. (a) Information propagation in the C matrix. The cell (3,4) considers the 4 cells above it as optional sources. (b) Computing the optimal indicator path for entry $(3, 4)$ given the choice of $(2, 1)$ as the source of the path, requiring $4 - 1 = 3$ consecutive indicators in row 3. (c) In a monotonic permutation matrix, every indicator has no indicators in its upper-right and lower-left submatrices (shadowed). Additionally, the indicators in every row are consecutive. (d) An example of mapping matrix M and its corresponding mapping matrix applying to CDF.

3.1. Runtime and Memory Complexities

As the auxiliary elements in the algorithm include T and C, the memory consumption is linear in the mapping matrix size, i.e. $O(mn)$. As for running time, in the most general case, running time is $O(n^2 m)$, as the rows are scanned once $(1 \ldots m)$, and for every row, a double loop $(1 \ldots n)$ of j and j' is performed.

In the next section, we show that when adopting the algorithm to work on cumulated density functions, the runtime can be further reduced to be as low as $O(mn)$.

3.2. Using the Algorithm on Cumulative Histograms

To extend the algorithm to cumulative histograms (CDFs), let us first define the *integration matrix* H to be a lower triangular matrix of ones, namely: $H(i, j) = 1$, if $i \geq j$, and 0 otherwise. To convert a set of normalized histograms (PDF set) into cumulative histograms (CDF), one may multiply it by H. Thus, we now change the objective function, and seek for the optimal M^* to satisfy:

$$M^* = \arg\min_{M \in \mathcal{S}} d(HMA, HB) \tag{3}$$

Since H is a full rank matrix we may write:

$$d(HMA, HB) = d(HMH^{-1}HA, HB) = d(\tilde{M}HA, HB)$$

It is easy to verify that $\tilde{M} = HMH^{-1}$ is a matrix of zeros except for a single indicator in each row with value 1. In the i^{th} row the indicator appears at column $\phi(i)$ where:

$$\phi(i) = \arg\max_{i'} \{\pi_M(i') \leq i\}$$

Meaning $\phi(i)$ returns the largest column index of the indicators in the first i rows in M (see example in Figure 2-d). Therefore, we can use an algorithm similar to the one used for histograms, but instead of working with histogram matrices A and B, we work with cumulative matrices HA and HB, respectively. Instead of searching for a *sequence* of indicators in M, we are looking for a *single* indicator in each row of $\tilde{M}$, still in a non-decreasing column index. Due to the monotonic non-decreasing property of the CDFs themselves, and due to the fact that we only choose a single indicator, we can run the algorithm even faster.

The cost calculation for finding $C(i,j)$ involves two parts: The cost caused by setting an indicator in column j, and the cost originated from the previous row, i.e. $\min_{j' \le j} C(i-1, j')$. The latter can now be calculated efficiently for each j by a single comparison. This reduces the runtime to be linear in the of number of elements in C, i.e $O(mn)$.

4. RESULTS

In order to demonstrate the advantage of GHM we evaluate the behavior of GHM at different number of histogram pairs and compare it to a couple of common alternatives. The first alternative is to average all histograms together and apply the standard HM to the histogram mean. This approach is commonly used for finding inter camera Brightness Transfer Function (BTF) [10]. The BTF is eventually a color map used to match images taken by two cameras. The second alternative is to compute the mapping for each pair of histograms independently, then average the mappings. This termed Mean Brightness Transfer Function (MBTF) [10]. The last alternative is the mapping method suggested by Porikli [11].

A source image of a static scene was taken using some "wrong" white balancing settings. The target image had the "correct" white balance, but the scene is slightly different. A ground-truth image is supplied for comparison. The setup is shown in the top row of Figure 3. We used the color histograms (each color band independently) to correct the colors of the source image, despite the changes in the scene. For comparison, the RMSE between the mapped and the reference images were calculated. All images of the scene are fairly aligned, so in order to get multiple histograms, the image was arbitrarily divided into square patches. This provides multiple histogram sources, and avoids the need for exact image registration. Patches or regions where the images are different are considered outliers, and we expect our method to properly calibrate the colors, despite those outliers.

The number of patches produced by the image division is a parameter that can be tuned. Setting it to 1, results in a single histogram pair, making GHM, MBTF and HM behave almost identically. As the number of patches is increased, the differences between the algorithms begin to emerge. Figure 3 shows a visual comparison between GHM, MBTF [10] and Porikli's method [11]. Note that MBTF deteriorates drastically when the number of patches increases, as small patches usually have sparse histograms, providing less information about the mapping outside their regions. Porikli's method demonstrates less sensitivity to the number of patches but, unfortunately, it produces poor looking results. The GHM results shows the advantage of dividing the image into multiple patches. It shows a perceptually better results when increasing the number of patches. Figure 4 shows RMSE between the reference and the mapped source vs. the number of patches. The graph is averaged over 6 sets of images, after dividing RMSE by the HM's result. It is demonstrated that GHM outperforms the other tested approaches.

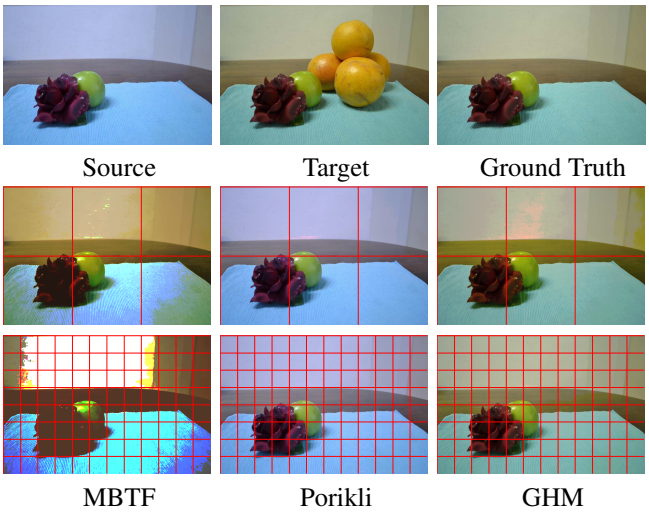

Fig. 3. Top: source, target and reference images. The goal is to map the colors of the source image, using the target image. Below it are the results of MBTF, Porikli, and GHM for 6 and 192 patches (marked with a red grid).

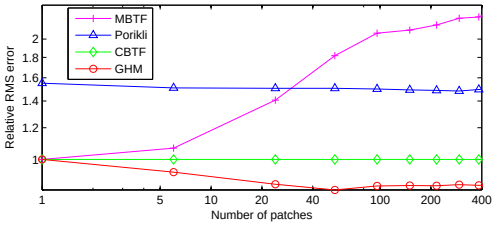

Fig. 4. RMSE relative to HM as a function of the number of patches. The error was averaged over 6 different scenes, such as shown in Figure 3.

5. CONCLUSIONS

We proposed a new algorithm, termed Generalized Histogram Matching (GHM), to find a monotonic mapping between two sets of images using their histograms. It extends Histogram Matching in three ways: (1) it can handle multiple histograms, (2) it can work with any additive distance metric (3) it can work either with PDFs or CDFs.

6. REFERENCES

[1] Y. Hel-Or, H. Hel-Or, and E. David, "Fast template matching in non-linear tone-mapped images," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1355–1362.

[2] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[3] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Computer Vision–ECCV 2006*, pp. 404–417, 2006.

[4] K.R. Castlman, *Digital Image Processing*, Prentice Hall, Englewood Cliffs,NJ, 1996.

[5] R.C. Gonzalez and R.E. Woods, *Digital Image Processing (3rd Edition)*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[6] Y. Rubner and C. Tomasi, *Perceptual Metrics for Image Database Navigation*, Kluwer Academic Publishers, Boston, Dec. 2000.

[7] M. Werman, S. Peleg, and A. Rosenfeld, "A distance metric for multidimensional histograms," *Computer Vision, Graphics, and Image Processing*, vol. 32, no. 3, pp. 328–336, 1985.

[8] O. Pele and M. Werman, "Fast and robust earth mover's distances," in *Computer vision, 2009 IEEE 12th international conference on*. IEEE, 2009, pp. 460–467.

[9] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *IJCV*, vol. 40, 2000.

[10] M. Shah O. Javed, K. Shafique, "Appearance modeling for tracking in multiple non-overlapping cameras," in *IEEE ICCV*, 2005, pp. 26–33.

[11] F. Porikli, "Inter-camera color calibration using cross-correlation model," in *ICIP*, 2003, pp. II:133–136.