Perceptually-Inspired and Edge-Directed Color Image Super-Resolution

Yu-Wing Tai  Wai-Shun Tong  Chi-Keung Tang
Vision and Graphics Group
The Hong Kong University of Science and Technology
{yuwing,cstws,cktang}@cs.ust.hk

Abstract

Inspired by multi-scale tensor voting, a computational framework for perceptual grouping and segmentation, we propose an edge-directed technique for color image super-resolution given a single low-resolution color image. Our multi-scale technique combines the advantages of edge-directed, reconstruction-based and learning-based methods, and is unique in two ways. First, we consider simultaneously all the three color channels in our multi-scale tensor voting framework to produce a multi-scale edge representation to guide the process of high-resolution color image reconstruction, which is subject to the back projection constraint. Fine details are inferred without noticeable blurry or ringing artifacts. Second, the inference of high-resolution curves is achieved by multi-scale tensor voting, using the dense voting field as an edge-preserving smoothness prior which is derived geometrically without any time-consuming learning procedure. Qualitative and quantitative results indicate that our method produces convincing results in complex test cases typically used by state-of-the-art image super-resolution techniques.

1. Introduction

The problem of single image super-resolution is under-constrained because many high resolution images may produce the same low resolution image. There is a fundamental limit on reconstruction-based super-resolution [2, 11]. In [15], approaches in single image super-resolution were categorized into three classes: interpolation-based, reconstruction-based and learning-based. Interpolation-based techniques (Figure 1(b)) are widely used for producing zoom-in images because of its simplicity. Besides that edge detection is an issue, the methods in [1, 10, 3] do not satisfy the reconstruction constraint leading to blurry results and aliasing artifacts along edges. Reconstruction-based approaches [9, 13, 2] satisfy their respective reconstruction constraints (Figure 1(c)–(d)). However, contour smoothness cannot be guaranteed and the reconstructed high-resolution edges are sometimes too sharp and look unnatural. Learning-based approaches [6, 8, 15, 4] learn
Figure 2. Overview of our approach. (a) input low-resolution image, (b) edge tokens with different degrees of “curve-ness” detected by tensor voting, where dense voting field robustly recovers and reinforces curve smoothness inherent in the input, (c) The “curve-ness” map, or reconstructed edge responses produced by tensor voting in high resolution, (d) final high-resolution image after enforcement of the reconstruction constraint. (Please see electronic version for better visualization.)

high frequency details from a large number of training images in the form of high-low resolution image patch pairs (Figure 1(e)). During reconstruction, image patches from the given low-resolution image are compared with the low-resolution image patches in training examples. A set of candidate high-resolution image patches is selected. Using the Markov chain model, the best high-frequency layer obtained is combined with the given low-resolution image using the reconstruction constraint to create the target high-resolution image.

The learning-based image hallucination produces arguably some of the best results to date [15]. The performance significantly relies on the quality of the priors used. The patch-based learning used in [15] is computationally expensive because a large image database is necessary to produce good results, so the authors used primal sketch priors for primitive layers (edges and junctions). The resolution that can be reconstructed is limited by the size of patches used, where a fixed number of discrete patch sizes and orientations were employed in order to speed up the learning process. Non-primitive layers (or non-edges) were not learned.

In this paper, we propose to use multi-scale tensor voting to address the problem of super-resolution from a single image. While most approaches operate on grey levels, our unified approach utilizes full color information in the image. Our method offers the advantages of learning-based, edge-directed and reconstruction-based methods, eliminates some of their respective disadvantages, and produces convincing results:

No time-consuming learning. Instead of learning primal sketch priors, tensor voting [12], which uses dense voting field, is used to vote for high-resolution curves at multiple scales by considering the interaction among different colors in the given low-resolution image. Dense voting field serves as an edge-preserving smoothness prior, and is geometrically derived but not learned by any computationally intensive process.

High-resolution curve extraction. Edge detection is an issue for edge-directed approach. In multi-scale tensor voting, edges are extracted at multiple scales. The multi-scale tensor voting extracts curves by enforcing the smoothness constraint and enhances insignificant edges that should lie on a salient curve. Using color information also enhances the estimation of sub-pixel location of edges. The extracted curves are used to guide the reconstruction process to avoid blurriness resulting by undesirable mixture of colors.

Enforcement of reconstruction constraint. The back projection algorithm is applied to preserve the overall resemblance between the low and high resolution images.

Figure 2 shows an overview of our approach. Dense voting field is used to extract multi-scale curves from the edge responses detected by multi-scale tensor voting. Then, high-resolution colors are reconstructed, followed by applying the back projection algorithm. In the following, we first provide a concise review of tensor voting and readers are referred to [12] for more details such as its robustness. Then, we describe a multi-scale tensor voting approach to color image super-resolution.

2. Multi-scale Tensor Voting for Perceptual Grouping

In this section, we first provide a concise review on tensor voting [12]. After developing the necessary terminologies, we describe how multi-scale and color analysis can be incorporated into the tensor voting framework.

2.1. Tensor voting

Tensor. Consider a 2D location, which is either a non-input or an input site. An input site can contain curve normal direction or otherwise. In order to uniformly encode an input site, a 2D symmetric tensor is used, which is given by the eigensystem where $\lambda_1 \geq \lambda_2 \geq 0$ are eigenvalues and $\hat{e}_1, \hat{e}_2$
are the corresponding eigenvectors:

\[(\lambda_1 - \lambda_2)(\hat{e}_1 \hat{e}_1^T) + \lambda_2(\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T)\]  

(1)

where \(\hat{e}_1 \hat{e}_1^T\) is a stick tensor, and \(\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T\) is a ball tensor.

An input site without any direction information is encoded into a ball tensor: \(\lambda_1 = \lambda_2 = 1\) and \(\hat{e}_1 = [0 1]^T\) and \(\hat{e}_2 = [1 0]^T\). An input site with curve normal \(\tilde{n}\) is encoded by a stick tensor: \(\lambda_1 = 1, \lambda_2 = 0\) and \(\hat{e}_1 = \tilde{n}\), and \(\hat{e}_2 \perp \hat{e}_1\).

**Voting fields.** To recover normal orientations and to enforce adaptive smoothness, a dense tensor voting field is used. The dense 2D voting field has two components: the stick component \(S\) and the ball component \(B\). The size of the voting field indicates the scale of analysis: the larger the scale, the more input sites are considered during the voting process.

**Voting process.** In the tensor voting process, each input (encoded) tensor uses the dense tensor field to cast tensor votes (stick and ball tensors) in its neighborhood with size determined by the scale of analysis. When each input tensor has collected and summed up all the votes cast to it, eigensystem decomposition is performed to interpret the collected votes.

**Interpretation.** Both the location and distribution of input sites have important implication. Points lying on a smooth curve receive many stick tensor votes pointing in the curve normal direction, while points lying inside a region receive votes uniformly from all directions. After decomposing the collected tensor votes at each input site into the corresponding eigensystem, we can define:

1. high “curve-ness”, which is indicated by large \((\lambda_1 \gg \lambda_2)\) with \(\hat{e}_1\) indicating curve normal direction. Second order votes are collected to infer curve-ness [12].
2. high “region-ness”, which is indicated by large \(\lambda_2\). First order votes are collected to infer region-ness [12].

**2.2. Perceptual grouping and tensor encoding for color images**

In the previous subsection, we mention that tensor voting determines “curve-ness” and “region-ness” of an input site (or token for short). In a color image, normal orientation for each token (pixel) is implicit and the token (pixel) distribution is uniform. In this subsection, we describe how a color image can be encoded into a set of tensor tokens so that we can apply tensor voting. The resulting tensors indicate the perceptual saliency “curve-ness” and “region-ness” at the pixel which are instrumental in producing good super-resolution results. The detailed steps are as follows:

1. Given a pixel \((x, y)\), the size of its neighborhood \(\mathcal{N}(x, y)\) (as determined by the given scale of analysis) we calculate the mean \(\mu\) and covariance matrix \(\Sigma\) of the Gaussian distribution using the pixel colors within \(\mathcal{N}(x, y)\). Here, we assume that the color distribution at each pixel can be adequately modeled by a single 3D Gaussian (for the R, G and B channels).

2. The “curve-ness” and “region-ness” of a pixel is measured by the spread of Gaussian distribution \(|\Sigma|\). If \(|\Sigma|\) is large, it is likely that the pixel is lying near the region boundary where a higher degree of color diversity is present. On the other hand, if \(|\Sigma|\) is small, it indicates that the pixel is lying in the interior of a smooth color region where the color transition is smooth.

Let \(|\Sigma|_{\text{min}}\) and \(|\Sigma|_{\text{max}}\) be the respective minimum and maximum \(|\Sigma|\). After the above color distribution analysis localized at each pixel, we encode each pixel into a tensor by

\[\beta S + (1 - \beta) B\]  

(2)

where \(\beta = (|\Sigma| - |\Sigma|_{\text{min}})/(|\Sigma|_{\text{max}} - |\Sigma|_{\text{min}})\), \(B\) is defined earlier, and \(S = \hat{e}_1 \hat{e}_1^T\) with the unit vector \(\hat{e}_1\) equals to

\[\frac{1}{N} \sum_{(x, y)} |I_{x'y'} - I_{xy}|^2 \left[\begin{array}{c} x' - x \\ y' - y \end{array}\right] \left[\begin{array}{c} x' - x \\ y' - y \end{array}\right]\]  

(3)

where \(I_{xy}\) and \(I_{x'y'}\) are the respective pixel colors at \((x, y)\) and \((x', y')\), \((x, y) \in \mathcal{N}(x, y)\) is a pixel within the neighborhood \(\mathcal{N}(x, y)\) of pixel \((x, y)\), \(N\) is the number of pixels in \(\mathcal{N}(x, y)\) and \(Z\) is a normalization factor to make the tensor unit. The three color channels are used simultaneously if the given low-resolution image is in color.

3. After encoding, tensor voting is executed to reinforce and refine the normal orientations of edges and to correct misaligned tokens as well.

Figure 3 shows the result of tensor encoding of a sample color image where an edge is present.
Figure 4. (a) Zoom-in of a region, (b) Subsampled representative tokens of the region. Based on the sampling theory, the representative tensor tokens are capable of reconstructing all other tokens faithfully by using a smooth reconstruction filter.

The above steps are executed at multiple scales (i.e., a set of discrete scales). For each token, the scale that gives the highest normalized “curve-ness”/“region-ness” consistent with of the local color distribution is taken as the optimal scale, $\sigma_{opt}$, at the token.

3. Multi-Scale Color Image Reconstruction

We describe in this section how multi-scale tensor voting infers the target high-resolution color image, by making use of the multi-scale features obtained in the previous section.

3.1. Subsampling

The purpose of subsampling is to extract a sparse set of representative tokens adequate to produce a faithful image reconstruction by applying tensor voting with the optimal scale at each representative token. This subsampling process also significantly speeds up the reconstruction process.

For simplicity, we do not consider cases where ambiguous features exist but consider the $\sigma_{opt}$ that corresponds to the highest normalized “curve-ness”/“region-ness” obtained in section 2.2. A small scale should used for the extracting fine details while a large scale for coarse details. Thus, the smaller $\sigma_{opt}$ is, the higher the sampling rate should be. Based on the Nyquist sampling theory [7], we use a grid-based approach for extracting a minimal set of representative tokens with the grid size proportional to $\sigma_{opt}$. Our grid-based approach is similar in spirit to quadtree decomposition, but is different because our grid size is derived from the optimal scale. These subsampled representative tokens will be placed at the corresponding locations in high-resolution image, that is, upsampled for color image reconstruction. Figure 4 shows the subsampling result.

3.2. Estimation of edge position at sub-pixel precision in the high-resolution image

The subsampled representative tokens with high “curve-ness” placed in the high-resolution domain (i.e. the upsampled tokens) suffer from the quantization error because of low-resolution image. Aliasing artifacts are very observable along edges in the reconstructed high-resolution image. It is therefore necessary to estimate the sub-pixel location of the upsampled curve tokens to alleviate the problem, and prevent mixing of colors across regions as well during the reconstruction of the high-resolution image. Note that this is not a problem for tokens with high “region-ness” because they represent homogeneous color or low-frequency region in local neighborhood where aliasing is not an issue.

Reconstruction-based approaches [9, 13, 2] estimate the sub-pixel location based on curve smoothness prior; learning based approach [6, 8, 15, 4] learn the sub-pixel location by finding patches in training example for which the down-sampled curve response is similar to the curve response in low-resolution image. However, these approaches do not take the full color information into account. The estimated sub-pixel location of the curve can, however, be very different from that of the ground-truth curve especially in learning-based approaches when there are not enough examples present in the training set. We believe that both color and curve smoothness should be considered simultaneously.

When a low-resolution image is under-sampled, the color along edges consist of a mixture of colors as a result of PSF of camera. For simplicity, we assume the mixture of colors come from the two sides of an edge and so adopt the matting equation [14] to model the color mixing proportion:

$$I(x, y) = \alpha F(x, y) + (1 - \alpha) B(x, y)$$

where $F(x, y)$ and $B(x, y)$ represent the respective colors on the two sides of an edge in the low-resolution image. We use Bayesian matting [5] to solve $F$, $B$ and $\alpha$. Recall that we have normal direction $\hat{e}_1$ of edge at each token estimated by tensor voting in section 2.2. The sub-pixel location of the curve token is given by $(x, y) + (2\alpha - 1)\hat{e}_1$ accordingly.

A pass of multi-scale tensor voting among the tokens with high “curve-ness” is followed to further refine the sub-pixel
locations by searching the maximum curve-ness along $\hat{e}_1$. Figure 5 shows the result of our sub-pixel location estimation based on both colors and curve smoothness.

### 3.3. “Curve-ness” map generation

The location-corrected and upsampled representative tokens with high “curve-ness” placed at high-resolution image domain consists of a sparse set of tokens. We need to reinforce the tensor tokens to produce a dense set of tokens for guiding the image reconstruction process. This can be achieved by casting stick tensor votes from the representative tokens to all quantized pixel locations in the high-resolution image domain, using the same tensor voting field at the optimal scale in the high-resolution which can be readily derived from the image magnification factor.

After multi-scale tensor voting, we produce a “curve-ness” map where each quantized pixel location contains a tensor with two pieces of important information: curve-ness and the corresponding curve normal direction (Figure 2(c)).

### 3.4. Color image reconstruction

Given the “curve-ness” map which is a dense tensor field of the same resolution as the high-resolution image, we reconstruct the target high-resolution image by generating the color at each pixel location $\langle x, y \rangle$ using the following PSF operation:

$$ C(x, y) = \frac{1}{W} \sum_{(x', y')} G : S(r)C(x', y') $$  \hspace{1cm} (5)

where $(x', y')$ is a location in the high-resolution image that has a corresponding pixel location we subsampled in the original low-resolution image, and $(x, y)$ is in the neighborhood of $(x, y)$. $G$ is a Gaussian with decay depending on the distance between $(x, y)$ and $(x', y')$. $S(r)$ is a Sigmoid function, where $r$ is calculated by accumulating the edge responses along the line joining $(x, y)$ and $(x', y')$ in the “curve-ness” map.

Figure 6 shows the effect of $S(r)$ which models the “sharpness” of the underlying curve according to the accumulated edge response. A smooth and continuous transition from sharp edges to soft edges as shown can be achieved without resorting to hard thresholds.

### 3.5. Enforcement of reconstruction constraint

After the above color image reconstruction, we obtain an intermediate result that does not satisfy the reconstruction constraint in general. The back projection algorithm introduced in [9] is an iterative gradient-based method that minimizes the reconstruction error:

$$ I_{H}^{t+1} = I_{H}^{t} + ((I_{H} * h) \downarrow - I_{L}) \uparrow \ast p $$  \hspace{1cm} (6)

where $I_{H}^{t}$ denotes the high-resolution image at iteration $t$, $I_{L}$ is the original input low-resolution image, $h$ is a convolution filter for down-sampling (down-sampling is denoted by |\downarrow| and up-sampling by |\uparrow|) and $p$ is the “backprojection” filter used for error correction, in which $h$ and $p$ are both equal to the PSF used for image reconstruction (Eqn. 5). Note that we use the same PSF operator $G : S(r)$ in backprojection process which implicitly considers both reprojection error and the high-resolution “curve-ness” map. In our case, the final result is obtained simply by using the intermediate result obtained in section 3.4 as the initial guess. A typical final high-resolution image is shown in Figure 2(d). In our experiments, we run for 30 iterations which produce good results without transferring high-frequency noise components.

### 4. Experimental Results and Comparison

In Figure 1, we have compared our approach with representative techniques for single image super-resolution. Our result for “Monarch” is free of ringing artifacts, preserves

<table>
<thead>
<tr>
<th></th>
<th>Bicubic</th>
<th>Back projection</th>
<th>Tensor voting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monarch</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMS</td>
<td>30.0</td>
<td>28.8</td>
<td>25.8</td>
</tr>
<tr>
<td>ERMS</td>
<td>34.8</td>
<td>31.5</td>
<td>27.6</td>
</tr>
<tr>
<td>Lena</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMS</td>
<td>18.4</td>
<td>16.9</td>
<td>15.2</td>
</tr>
<tr>
<td>ERMS</td>
<td>18.6</td>
<td>19.2</td>
<td>17.8</td>
</tr>
<tr>
<td>Girl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMS</td>
<td>11.3</td>
<td>9.5</td>
<td>7.4</td>
</tr>
<tr>
<td>ERMS</td>
<td>14.5</td>
<td>10.2</td>
<td>8.6</td>
</tr>
<tr>
<td>Head</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMS</td>
<td>21.0</td>
<td>17.0</td>
<td>16.3</td>
</tr>
<tr>
<td>ERMS</td>
<td>14.6</td>
<td>18.9</td>
<td>17.5</td>
</tr>
</tbody>
</table>

Table 1: RMS and Edge RMS pixel error for different approaches. We compare the Edge RMS to show the ability of our approach in preserving edge features.
the necessary contour smoothness while maintaining the appropriate amount of image sharpness (See the zoom-in region for better comparison).

In this section, we tested our approach using other standard images. The input low-resolution image is produced by blurring and downsampling the available high-resolution image (the source images “Girl” and “Tomato” are courtesy of [15]), which is used as the ground truth for qualitative and quantitative comparison. Please also refer to the electronic version of the paper.

**Comparison with interpolation-based approaches.** We compare our approach with bicubic interpolation and back-projection on “CVPR Logo” as a text example in Figure 7. As shown in the zoom-in region, the curves extracted by our approach are smoothest while the most crisp, whereas blurry and aliasing artifacts are present in their results. We further compare the results on the standard “Lena” image. Figure 8 show the results produced by bicubic interpolation, sharpened bicubic interpolation (using the “unsharp mask” in Adobe Photoshop with the default parameters),
the backprojection, and tensor voting. The bicubic result is the smoothest among all the results. Sharpened bicubic and backprojection methods introduce strong ringing effects, especially along the salient image contours (e.g. the brim of the hat). Our result preserves the appropriate amount of sharpness and smoothness because the reconstruction is guided by the “curve-ness” map.

Comparison with reconstruction-based approaches. While the RMS and the edge RMS pixel errors (Table 1) of our approach are better than those produced by backprojection [9], our results (shown in Figures 1(g), 8(e) and 9(c)) have less observable ringing artifacts than the results obtained by backprojection, which optimizes for the resemblance to the low-resolution image. Note the ringing artifacts near the feeler and the top of the monarch in Figure 1(d), the brim of the hat of “Lena” 8(d) and the silhouette of the face in Figure 9(a) as well. The level set approach produces unnaturally very sharp edges while it fails to extract the feeler of monarch (Figure 1(e)). The cubic spline interpolation, on the other hand, produces a blurry result compared with our approach (Figure 10(c) and (f)).

Comparison with learning-based approaches. Compared with image hallucination [15], our results preserve fine details better in some standard cases. We do not blur the feeler of the monarch (Figure 1(f) and (g)). The details such as the hairs, the eyelids, the mouth and the mole in “Girl” are preserved faithfully due to our edge-directed approach (Figure 9(b) and (c)). In comparison with Freeman et al.’s method [6] and neighbor embedding [4], our results preserve more color and texture details (Figure 10). Because no reconstruction constraint is enforced in their approaches, the learnt edges might be very different from the edges of the original image. Also, the results of learning-based approaches strongly depend on the amount and quality of training examples being used.

Quantitative comparison. Table 1 compares the RMS and the edge RMS pixel errors on the respective cases of various approaches. Our tensor voting approach performs better than some of the representative interpolation-based and reconstruction-based methods. Our RMS errors and ERMS errors for “Monarch” and “Girl” are comparable to those produced by image hallucination [15] ([26.9 29.4] for “Monarch” and [7.1 9.1] for “Girl” respectively) while the visual appearances of our results are arguably better. Because no reconstruction constraint is considered in Freeman et al. [6] and neighbor embedding [4], our RMS and ERMS errors are better in the cases of “Head” ([24.3 30.0] and [22.2 27.5] respectively), and our results are visually more satisfactory. The tensor voting approach, however, does not require any learning. All high-resolution details presented in this paper are produced and guided by multiscale tensor voting.

Running times. We run our experiments on a notebook computer with Intel Pentium M 1.40 GHz processor and 1 GB RAM. The running time of our approach depends on factors such as image resolution, magnification factor, and the number of salient edges in the image. Table 2 summarizes the running times. Figure 11 shows three additional results.
5. Summary and Conclusion

In this paper we present a unified approach by multi-scale tensor voting for inferring a high-resolution color image from a single low-resolution color image. Our method is inspired by perceptual grouping where each color pixel is grouped as a region or an edge token with appropriate degrees of confidence we term as “curve-ness” or “region-ness”. This is achieved by multi-scale tensor voting, after encoding all color pixels into tensors based on the local color distribution at each pixel. Our edge-directed reconstruction approach makes use of the dense voting field, which is geometrically derived but not learned [12], to hallucinate high-resolution curves by enforcing a smoothness constraint. These curves are used to guide the color reconstruction process so that fine details are inferred at a higher resolution (subject to a certain fundamental limit). The experimental section shows that our approach offers the advantages given by edge-directed, reconstruction-based, and learning-based approaches within the same tensor voting framework, and that better or comparable results are obtained for typical and complex images.

References