The Visual Computer (2019) 35:565–577
https://doi.org/10.1007/s00371-018-1485-y

ORIGINAL ARTICLE

Fast example searching for input-adaptive data-driven dehazing with Gaussian process regression

Xin Fan[1] · Xianxuan Tang[2] · Minjun Hou[2] · Zhongxuan Luo[2]

Published online: 20 February 2018

Abstract

Recently, data-driven approaches are prevailing in low-level image processing including single image dehazing. The performance of these methods can behave better when the learning process adapts to the input. This input-adaptive training demands efficiently selecting optimal examples for the input from a large training set. In this paper, we address the issue of input-specific example searching and propose a fast searching strategy on vast image examples to learn a more accurate Gaussian process (GP) regressor for single image dehazing. The GP regression learnt from these optimal examples is able to produce the transmission prediction with lower variance and thus renders high robustness. Extensive experiments on hazy images at various haze levels demonstrate the effectiveness of the proposed example searching compared with the state-of-the-art data-driven dehazing methods.

Keywords Image dehazing · Example searching · Gaussian process regression · Input adaptive

1 Introduction

There has been growing interest in data-driven approaches to low-level image processing including image super-resolution [33], denoising [37], and dehazing [35]. These approaches generalize those low-level processes specific to applications (tasks) to learn *generic* models from a large number of examples designated to the tasks. For instance, the algorithms for super-resolution [20], denoising [25], and dehazing [12] learn similar Gaussian process (GP) models, and Markov random fields (MRFs) are common in the processes for super-resolution [15] and dehazing [19]. But different types of training example pairs are fed to these common models for different tasks. Naturally, data are turning out to be the focus of different data-driven image processing techniques, but neglected in many studies. In this paper, we address the issue of seeking optimal training examples that yield the improved performance on a given input.

Recent works typically learn direct mappings from example pairs of a degraded input to its desired output. Zoran et al. [40] propose a framework to learn the mid-level visual properties of an image and perform the estimation of reflectance, shading, and depth. Burger et al. [3] use pairs of noisy and clean image patches as training data to learn the parameters of a multilayer perceptron (MLP) model for denoising. Tang et al. [35] learn the relationship between input features and transmission using the random forrest regression for dehazing, while we model this relationship with Gaussian processes [12]. Zhu et al. [39] learn the parameters of a linear model and then recover scene depth information for transmission estimation. These learning-based methods are able to yield superior performance when abundant training pairs are available.

“More data beats a clever algorithm” [8]. The quantity of data might be not an issue, but their *quality* is really critical in this “big data” era. Regressors would behave better when the regression process adapts to the input. Schmidt et al. [30] adaptively train a discriminative regressor for deblurring by minimizing a loss function upon the training set. Selecting sparse inducing points and manually re-labeling the training examples can also improve the performance of regression models [31]. In our previous work [12], a simple screening step with trained support vector machines (SVM) is able to upgrade the performance of GP regression for dehazing.

✉ Xin Fan
xin.fan@ieee.org

[1] DUT-RU International School of Information Science and Engineering, Dalian University of Technology, Dalian, China

[2] School of Software, Dalian University of Technology, Dalian, China

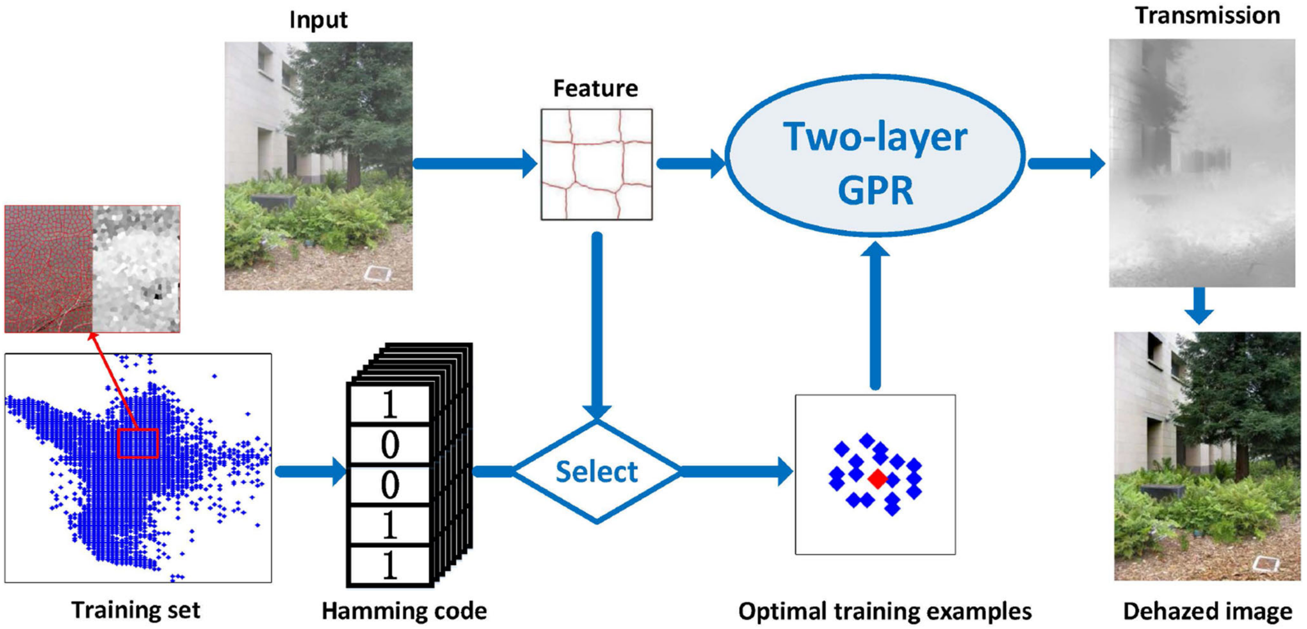

Fig. 1 Algorithm overview. We first search optimal training examples for an input synthetic hazy image to train a Gaussian process (GP) regressor and thus estimate a more accurate transmission map for dehazing by applying the trained input-adaptive GP regressor

Nevertheless, it is still an open issue to efficiently find the training examples from a large volume of data upon which more accurate regressors can be learnt for low-level processing of a given input.

In this paper, we propose a searching strategy on image examples (including those collected from the Web) for learning a more accurate GP regressor for image dehazing. We prove that the training examples neighboring the input are able to train a GP regressor with lower predictive variance. We leverage the Hamming embedding [24] to efficiently search these examples neighboring to the input. Our strategy, which takes the modality-specific learning to constitute a more accurate regressor for the given modality (input), is also validated by recent cognitive studies [17]. Figure 1 shows the overview of the proposed method for a synthetic hazy image. The resultant image is quite close to the original haze-free by recovering image textural details as well as the chromatic information. More experimental results validate the effectiveness of our method in Sect. 5.

2 Related work

In this section, we review existing dehazing methods and recent advances that improve the GP regression.

2.1 Image dehazing

Hazy images with low visibility influence both human perception and computer vision. Pioneering dehazing works in the past decade typically rely on prior modeling of the physical image formation process. Fattal [13] proposed a refined image formation model that includes the surface shading and assumed that the transmission is uncorrelated with the surface shading. He et al. [22] estimated the transmission map of hazy image under the assumption of dark channel prior (DCP) that the local minimum of RGB channels in a haze-free image is close to zero. These image priors are applicable to certain kind of images, but not to generic real-world images. They work well in the scenarios following those assumptions, otherwise fail.

Recently, researchers have resorted to haze removal from a learning perspective. Gibson et al. developed a learning framework for haze removal using synthesized hazy images with known fog and depth [18]. Tang et al. investigated haze-relevant features and applied random forests for transmission learning [35], and we developed a two-layer GP regression to generate more smoothing transmission estimation [9]. Zhu et al. used a supervised learning method to train the parameters of a linear model [39]. There also exist dehazing methods using deep networks. Ren et al. employed a multiscale convolutional neural networks (CNN) to estimate transmission maps for hazy images [27], while Cai et al. built an end-to-end system from images (instead of features) to transmissions [4]. These works, focusing on learning the relationship between features (or images) and transmissions, collect many synthetic hazy images to constitute a fixed training set for all input testing images. Also, the fixed set with synthetic images can hardly cover great variations of real-world hazy images. It still remains unresolved to find 'optimal' training examples, the most critical issue for data-driven processing.

2.2 Improvements on Gaussian process regression

Gaussian processes regression model is a simple and flexible model and also a powerful tool in many areas. Significant efforts have been invested to the Gaussian process regression, yielding improvements from various aspects. Miguel et al. introduced a sparse Gaussian process regression model and sparsified the spectral representation of the GP, which makes the regression model more simple and efficient [26]. In a recent study, Kwon et al. improved the quality of restoring degraded images by learning a semi-local GP regression model [25]. Instead of training a single GP model on a large data set, they constructed a set of sparse models to perform the prediction at each testing point. Cao et al. introduced an efficient optimization algorithm for GP regression and achieved the joint selection of inducing points and estimation of GP hyper-parameters by optimizing a single objective [5]. These improvements for GP regression greatly reduce the time complexity. Unfortunately, few methods increase the precision from the aspect of choosing training examples. In this study, we develop a systematic selection process to search the appropriate training set for a given input to improve the accuracy of GP regression.

3 Training example searching

Training examples have such great impacts on regression methods that a training process with examples adapting to given inputs may significantly improve the performance. In this study, we propose an efficient searching strategy to select optimal training examples from a vast of data points with various sources (the synthesized or Web). This section gives the proof on how the searching improves the precision of GP regression and presents the fast algorithm derived from the Hamming embedding.

3.1 Optimal examples for Gaussian process regression

3.1.1 Prediction distribution

The GP regression is not only able to learn a mapping from the input to the target with a set of training examples, but also to provide the probability distribution for predicting the target given a new input. This predictive probability gives the estimates of the target as well as the prediction precision dependent on training examples. We focus on the derivation of the distribution relating the precision with training examples.

Similar to [12], we build the nonlinear mappings from input features $\mathbf{f}$ to target transmission t with GP regression. The transmission t is a function $\Phi(\mathbf{f})$ of the input $\mathbf{f}$ with the additive noise ε amenable to a Gaussian distribution $N(0, \sigma_\varepsilon^2)$, expressed as:

$$t = \Phi(\mathbf{f}) + \varepsilon. \tag{1}$$

The covariance matrix of the marginal distribution for the target t is determined by the Gram matrix $\mathbf{G}$:

$$\mathbf{G}(\mathbf{f}_i, \mathbf{f}_j) = k(\mathbf{f}_i, \mathbf{f}_j) + \sigma_n^2 \delta(\mathbf{f}_i, \mathbf{f}_j), \tag{2}$$

where $\delta(\cdot)$ is the Kronecker delta function, $\mathbf{f}_i$ and $\mathbf{f}_j$, respectively, represent two features in the input features f, and $k(\mathbf{f}_i, \mathbf{f}_j)$ is a kernel function of $\mathbf{f}_i$ and $\mathbf{f}_j$. We take the squared exponential as the kernel:

$$k(\mathbf{f}_i, \mathbf{f}_j) = \sigma_f^2 \exp\left[\frac{-(\mathbf{f}_i - \mathbf{f}_j)^2}{2l^2}\right], \tag{3}$$

where σ_f^2 is the maximal allowable covariance and l is the length parameter.

We focus on the prediction of target t^* from a new input $\mathbf{f}^*$ given N_t training inputs $\mathbf{F}_{N_t} = [\mathbf{f}_1, \ldots, \mathbf{f}_{N_t}]$ and the corresponding observations $\mathbf{t}_{N_t} = [t_1, \ldots, t_{N_t}]^T$. The GP regression provides the conditional probability density $p(t^* \mid \mathbf{t}_{N_t})$ following a Gaussian distribution with mean $m(\mathbf{f}^*)$ and variance $\sigma^2(\mathbf{f}^*)$:

$$p(t^* \left| \mathbf{t}_{N_t} \right.) \sim N(m(\mathbf{f}^*), \sigma^2(\mathbf{f}^*)). \tag{4}$$

According to the GP regression process, the mean of the $\mathbf{t}_{N_t}$'s distribution is taken as the estimate of the predicted transmission t^*, expressed as

$$m(\mathbf{f}^*) = \mathbf{k}_* \mathbf{G}^{-1} \mathbf{t}_{N_t}, \tag{5}$$

where the N-dimensional vector $\mathbf{k}_*$ is a function of the input $\mathbf{f}^*$ as $\mathbf{k}_* = [k(\mathbf{f}^*, \mathbf{f}_1), \ldots, k(\mathbf{f}^*, \mathbf{f}_{N_t})]$, and $\mathbf{G}$ is a $N_t \times N_t$ kernel matrix of training sample. The variance $\sigma^2(\mathbf{f}^*)$ reflects the prediction precision, deduced as

$$\sigma^2(\mathbf{f}^*) = k(\mathbf{f}^*, \mathbf{f}^*) + \sigma_n^2 - \mathbf{k}_* \mathbf{G}^{-1} \mathbf{k}_*^T. \tag{6}$$

Refer to [2] for the detailed training process of hyper-parameters and the solving process of the mean $m(\mathbf{f}^*)$ and the variance $\sigma^2(\mathbf{f}^*)$.

3.1.2 Optimal training examples

The $\mathbf{k}_*$ term in Eq. (6) relates the predictive variance for a given input feature $\mathbf{f}^*$ with the relationship between $\mathbf{f}^*$

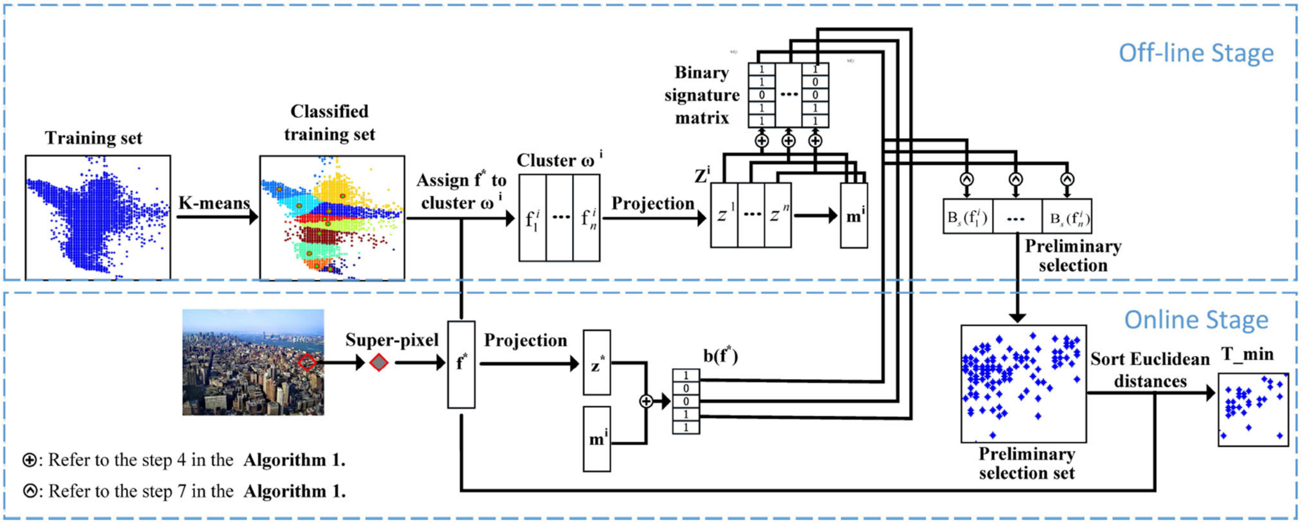

Fig. 2 The process of the online searching. For every super-pixel in the input image, we assign it to the corresponding cluster. Then, we represent the test feature and the features in the training set by binary signatures. Thus, the preliminary selection data can be got by calculating the binary scores. Finally, we sort the Euclidean distances between the test feature vector $\mathbf{f}^*$ and the feature vectors in the preliminary selection set to achieve the final training set T_min

and available training examples. We prove that the closest training examples to $\mathbf{f}^*$ train a GP regressor yielding lower predictive variance for the input $\mathbf{f}^*$.

In order to investigate the effect of a training pair $\mathbf{f}_i$ and t_i on the prediction, we peer into the variance of the predictive distribution $p(t^* |t_i)$ conditional on the observed target t_i of the input $\mathbf{f}_i$:

$$\sigma^2(t^* |t_i) = k(\mathbf{f}^*, \mathbf{f}^*) + \sigma_n^2 - k(\mathbf{f}_i, \mathbf{f}^*)\mathbf{G}^{-1}k(\mathbf{f}_i, \mathbf{f}^*)^T. \tag{7}$$

The last term in (7) reveals that the predictive variance for the GP regression depends on the connection between the training example $\mathbf{f}_i$ and given input $\mathbf{f}^*$. Substituting (2) and (3) into (7), we make the dependency more evident:

$$\sigma^2(t^* |t_i) = \sigma_f^2 + \sigma_n^2 - \frac{k(\mathbf{f}_i, \mathbf{f}^*)^2}{\sigma_f^2 + \sigma_n^2}. \tag{8}$$

The term $k(\mathbf{f}_i, \mathbf{f}^*)$ determines the predictive variance given the learned hyper-parameters σ_f^2 and σ_n^2. As shown in (3), the function is monotonic and inversely proportional to the exponential of the Euclidean distance between $\mathbf{f}_i$ and $\mathbf{f}^*$. Hence, the closest training examples to the input of interest $\mathbf{f}^*$ are able to output the prediction with a lower variance. It is possible to apply this dependency for efficient online training when data points sequentially arrive [25], while herein we dedicate to devise a fast strategy to localize these “optimal” ones from a collection of examples for accurate prediction.

3.2 Fast searching of optimal examples

As shown in (8), the Euclidean distance between the testing input and training examples weighs the prediction accuracy of GP regression. Unfortunately, it would be notoriously time-consuming if we directly calculate all Euclidean distances and find the closest ones at each time when predicting a new input. One straightforward strategy is to construct a kd-tree of training examples for acceleration [32,36]. The hamming embedding, adopted in large-scale visual retrieval [24], provides a more informative representation for distance pairs between feature vectors by binary signatures. These signatures do not only reflect rich contextual information, but also have extremely low computational loads and few memory usage. The searching process divides into off-line and online stages given below.

3.2.1 Off-line stage

The off-line process constitutes an efficient index structure for all available training examples in order to accelerate the online searching of optimal examples for the input. As shown in Fig. 2, we first collect a training set containing both synthetic and real-world images. Subsequently, we cluster examples in the training set and generate binary signatures for these examples.

We construct a training set with both synthetic and real-world images. Our previous transmission model in [11] is able to generate hazy images for natural scenes from original sharp images and their corresponding depth maps [29].

For real-world images, we apply the screening process in [12] to categorize natural images into three levels of haze and perform three traditional dehazing methods, [22,34], and [13] to dense, moderate, and light hazy images, respectively. Refer to [12] for the justification of this haze generation process upon haze levels. The target transmission maps are available for real-world images as a common by-product of these dehazing methods. Hereafter, we have the super-pixel features (detailed in the next section) and their corresponding transmissions as the target labels for training.

We reduce the dimensionality of training features upon their closeness for the sake of efficiency. The classical k-means algorithm groups the training feature vectors of super-pixels into ω clusters. Subsequently, we generate a $d_f \times d_f$ matrix (d_f is the dimensionality of feature vectors) of i.i.d. random values from a Gaussian distribution $N(0, 1)$ and apply the QR decomposition to the matrix, yielding orthogonal bases. The first d rows of the resultant orthogonal matrix are taken as the projection matrix $\mathbf{Q}_d$ for the dimensionality reduction. Multiplying the matrix $\mathbf{Q}_d$ with the training feature matrix $\mathbf{F}_n^i = [\mathbf{f}_1^i, \ldots, \mathbf{f}_n^i]$ of the i-th cluster ω_i, where n is the number of features in the cluster, we have the new training feature matrix $\mathbf{Z}^i$ for ω_i. We then compute the median value of each row in $\mathbf{Z}^i$ and obtain a median vector $\mathbf{m}^i = [m_1^i, \ldots, m_d^i]^T$ for the cluster ω_i. This median vector facilitates the fast localization of the cluster and generation of binary signatures in the online stage for example searching.

3.2.2 Online stage

In the online process, we generate the binary signature of the input and then search the examples close to the input through the binary index of the training set generated in the off-line process.

Given a new input image, we assign its feature $\mathbf{f}^*$ to the cluster ω^j with the closest centroid and then project the feature to a vector $\mathbf{z}^* = [z_1^*, \ldots, z_d^*]^T$ by $\mathbf{Q}_d$. The bit b_k is set to one if the k-th component of the projected vector z_k^* is larger than the corresponding median value of the cluster m_k^j, otherwise to zero. Hence, we generate the binary signature $\mathbf{b}(\mathbf{f}^*) = [b_1(\mathbf{f}^*), \ldots, b_d(\mathbf{f}^*)]^T$ for the input. The similarity between this binary feature and training features reflects how close the input feature is to the training ones. More importantly, this similarity score B_s can be efficiently evaluated by applying the binary exclusive operator to $\mathbf{b}(\mathbf{f}^*)$ against those binary features in the cluster. The online searching process is illustrated in Fig. 2, and the algorithm is summarized in Algorithm 1.

Algorithm 1 Online searching procedure

```
Assign the input feature $\mathbf{f}^*$ to the corresponding cluster $\omega^j$.
$\mathbf{z}^* \leftarrow \mathbf{Q}_d\mathbf{f}^*$
for $k = 1 \rightarrow d$ do
    $b_k(\mathbf{f}^*) = \begin{cases} 1, & \text{if } z_k^* > m_k^j \\ 0, & \text{otherwise} \end{cases}$.
end for
for $n = 1 \rightarrow N_f^j$, the number of features in $\omega^j$ do
    $B_s(\mathbf{f}_n^j) \leftarrow \sum_{k=1}^{d} \delta(b_k(\mathbf{f}^*), b_k(\mathbf{f}_n^j))$, $\delta$ is the Kronecker delta function
end for
```

Those feature vectors having the similarity score above a threshold b_t in the cluster ω^j are chosen as the candidate training set. The Hamming embedding algorithm significantly reduces the memory and time expenses since similarity evaluation on binary signatures has a negligible computational load. Subsequently, we sort the Euclidean distances between the input feature $\mathbf{f}^*$ and the feature vectors in the candidate set and take the N_t feature vectors with the closest distances as the final training set T_min. The similarity score on binary features efficiently localizes the candidate set, while the closest training features are picked upon the Euclidean distances between original features. This strategy balances the efficiency and accuracy.

4 Regression model for dehazing

Researchers have devoted great efforts to haze removal from a data-driven perspective in recent years. These data-driven methods typically learn image priors from training examples and achieve better performance than the classical methods upon physical models. In this study, we employ the two-layer Gaussian process regression to learn the mapping from features to transmissions as our previous work [12]. For the completeness of this paper, we sketch the regression model for which we search optimal training examples.

4.1 Hazy image formation model

The widely used formation model of hazy image [22,35] is as follows:

$$I(p_i) = J(p_i)t(p_i) + A(1 - t(p_i)), \tag{9}$$

where p_i is a pixel, I is the hazy image, J is the haze-free image of I, A is the atmospheric light, and $t(p_i)$ is the medium transmission of p_i that characterizes the portion of the light reaching the camera.

We slightly modify the transmission model to refine the transmission [11]. The refined transmission can be derived as:

$$t_r(p_i) = t_e(p_i)^{1 - Dvis_e/Dvis_r}, \tag{10}$$

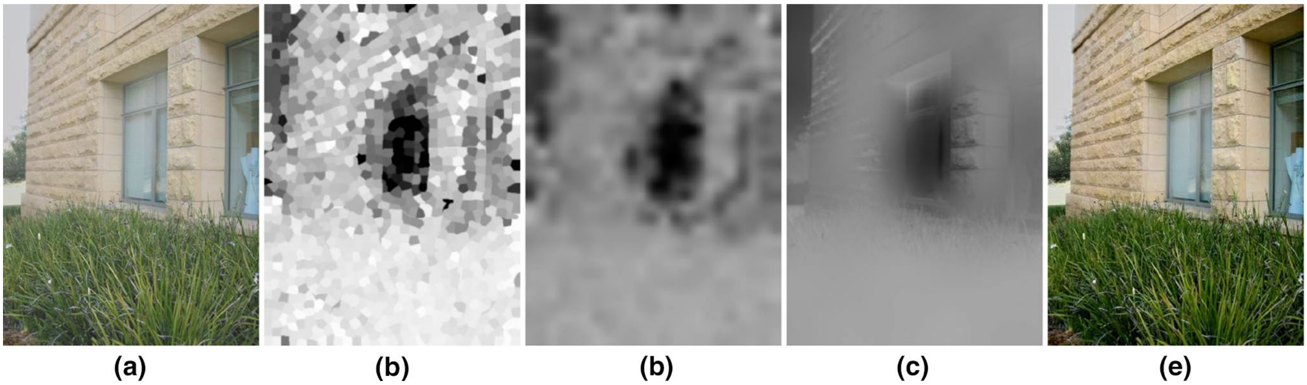

Fig. 3 Dehazing with GPR model. **a** A hazy image. **b** The transmission map estimated by the first GPR layer. **c** The transmission map estimated by the second GPR layer. **d** Final transmission map refined by guided filtering. **e** Our final dehazing result

where the $t_e(p_i)$ is the original estimated transmission and the $t_r(p_i)$ is the refined transmission. The two parameters $Dvis_e$ and $Dvis_r$ are the maximum visibility values for the original and desired images, respectively. By tuning the ratio of the two parameters, users can control the degree of haze in the resultant image.

4.2 Multiscale feature vector

Haze-relevant features form the input vector for regression. We use the hue disparity [6] between the hazy and its semi-inverse images as one feature partially attributing to its ability to detect haze [1]. As shown in previous studies on image dehazing, the dark channel [22], local maximum contrast, and saturation are highly correlated with the amount of haze. All these quantities vary with the local window size. Thus, we generate these values across various scales as features. The Gabor feature [6] represents the texture of image, and its value has a notable change in haze region. We convolve the input hazy image with a set of Gabor filters and calculate the Gabor features from the filtered image. Finally, the input feature vector includes the hue disparity, dark channel, local maximum contrast, saturation, and Gabor features.

4.3 Regression models

We employed a two-layer GPR model to learn transmissions of hazy image. The first layer takes the feature vector as the input and outputs the preliminary transmission. The second layer smoothes the transmissions predicted by the first layer and preserves the consistency of image structures.

4.3.1 The first layer of GPR

For the first layer, we take the average feature vector within a super-pixel [28] S_i as the input $\mathbf{f}_i$ and the average transmission within S_i as the target output since pixels in local region with similar structural contexts tend to have similar amount of haze. The $\mathbf{f}_i$ can be expressed as:

$$\mathbf{f}_i = \frac{1}{|s|} \sum_{p_i \in S_i} \tilde{\mathbf{f}}(p_i), \tag{11}$$

where s is the number of pixels in S_i and $\tilde{\mathbf{f}}(p_i)$ is the multi-scale feature vector of the pixel p_i. The process of obtaining the optimal training examples is described in Sect. 3.2.

Given an input feature vector $\mathbf{f}^*$ of an image to be dehazed, we can obtain the conditional probability of the target transmission t_f^* by the trained GPR. The conditional probability is a Gaussian distribution:

$$p(t_f^* | \mathbf{T}) \sim N(m(t_f^*), \sigma^2(t_f^*)), \tag{12}$$

where $\mathbf{T}$ are the transmissions of training data, $m(t_f^*)$ and $\sigma^2(t_f^*)$ are the mean and variance of this distribution, respectively, and the values of $m(t_f^*)$ and $\sigma^2(t_f^*)$ are taken as the predicted transmission $\mathbf{f}^*$ and its error, respectively. One assumption of our algorithms is that the pixels within a super-pixel present an identical depth, and thus a same transmission as the transmission is related to the depth:

$$t = e^{-\lambda d}, \tag{13}$$

where λ is a hyper-parameter that is independent of the transmission t and depth d. Therefore, heterogeneous pixels, i.e., those with different depths, given by the super-pixel segmentation may produce inaccurate transmission estimation. Fortunately, pixels in a super-pixel are more likely to share common structural contexts than those of a regular patch. Consequently, regressions upon super-pixels in our approach perform better than traditional path-wise regressions.

According to the predicted transmission of every super-pixel, we can obtain the transmission map of hazy image as shown in Fig. 3b. The transmission map can roughly reflect the depth and global structure of the image, but exhibits local disparity across super-pixels.

4.3.2 The second layer of GPR

The second layer builds connections between latent variables similar to the Markov random fields (MRFs) in [16,38] without any iterative energy optimization or inference process. The target of the second GP regressor is the averaged transmission within current super-pixel S_i, and the input $\tilde{\mathbf{t}}_i$ is the collection of its eight neighbors $N_e(S_i)$, where the $\tilde{\mathbf{t}}_i$ can be expressed as:

$$\tilde{\mathbf{t}}_i = [t(S_1), \ldots, t(S_j), \ldots, t(S_8)]_{S_j \in N_e(S_i)}. \tag{14}$$

The process of obtaining the training transmissions is the same as the first layer. Since a super-pixel does not necessarily share a boundary with eight adjacent neighbors as a pixel does, we take the eight neighbors nearest to the current super-pixel as the input.

In the prediction, we take the transmission of an input super-pixel S_i^* as the target $\tilde{\mathbf{t}}^*$ and the transmissions of its eight neighbors estimated by the first layer as the input vector. Then, the conditional probability of predicted transmission $\tilde{\mathbf{t}}^*$ follows a Gaussian distribution. Similarly, the mean of the Gaussian distribution is taken as the predicted transmission of S_i^*. The second layer maintains the consistency of image structures and attenuates the local disparity in the output of the first layer. As shown in Fig. 3c, the transmission map estimated by the second layer imposes the local smoothness to the transmission map of the first layer. We apply the guided filtering [21] to achieve a further refined transmission map for the haze removal and then restore the sharp image using the final transmission map and (9) as shown in Fig. 3e.

5 Experimental results and analysis

In this section, we compare the regression using the proposed example searching with the previous two-layer GP regression [9], where all images available in a data set were used for training, in order to verify the effectiveness of the searching strategy. As for the hyper-parameters in the online stage, we set $\omega = 10$, $d = 16$, and $b_t = 15$, which are fixed for a wide variety of input images while training features adapt to a specific input. To avoid unstable behavior of the GP regression, we take the chosen number of training features $N_t = 10$. The input feature vector has 37 dimensions ($d_f = 37$) including the hue disparity, dark channel (four scales), local maximum contrast (four scales), saturation (four scales), and Gabor features (three scales and eight directions).

Table 1 Execution time comparisons (s)

Method	Direct searching	kd-tree	Hamming bedding
Time	965.17	184.50	28.65

We also demonstrate the superior performance of our input-adaptive dehazing with example searching by comparing with four recently developed dehazing algorithms [22, 27,35,39]. As a nontrivial by-product, we collect different kinds of testing hazy images including people, buildings, landscape, etc., and categorize them upon the hazy degree for performance evaluation of dehazing algorithms.[1]

5.1 Execution time

In this paper, we use the hamming embedding to accelerate the example searching process. The hamming embedding converts real feature vectors of training super-pixels into binary signatures and applies binary operators for similarity comparisons during the searching process. These binary signatures and operators have negligible computational costs and memory storage, resulting in time and space efficient example searching. Table 1 lists the averaged execution time of directly exhaustive searching, searching with a kd-tree structure [36], and the proposed strategy on all available training features of super-pixels. If we directly calculate all the Euclidean distances between the input and training examples, the selection process costs as high as 965.17 s. The hamming embedding significantly reduces the time consumption to 20.67 s, which is acceptable in practice. Also, we use the directly exhaustive searching as the baseline to calculate the accuracy of the accelerating searching techniques. The accuracy of finding the optimal examples for the hamming embedding is 85%, higher than that of the kd-tree, 80%. The hamming embedding outperforms the kd-tree in terms of both accuracy and efficiency.

5.2 Comparisons using different training data sets

As we shown in Sect. 3.1, the variance of predicted transmission is directly related to the similarity between the training feature and the input. Herein, we compare the variances using three different training sets, i.e., T_max, T_min, and T_mid. The set T_min includes ten training examples having the lowest Euclidean distances to the input, while T_max and T_mid consist of those with the ten largest and median distances, respectively. These three sets train the GP regression model and then dehaze the input image with the trained regression. The variance of the estimated transmission for every

[1] All resultant images for these comparisons and testing hazy images are available at https://github.com/dlut-dimt/TVCJ.

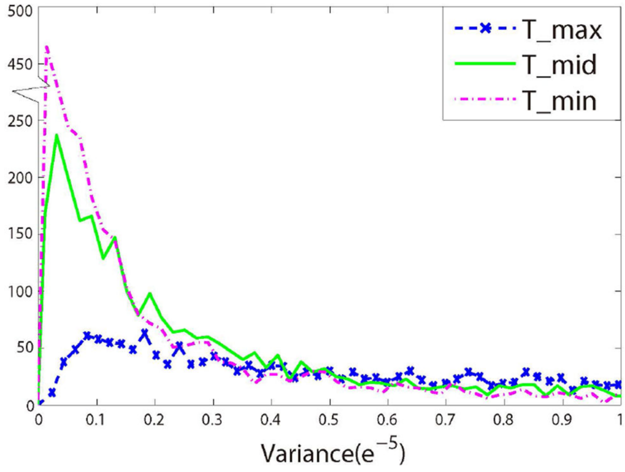

Fig. 4 Variance distributions of GPR by T_max,T_min, and T_mid training sets

Table 2 Mean variances ($\times 10^{-6}$) for 3 hazy images

Image	T_max	T_mid	T_min
realhaze6	54.24	36.68	2.65
realhaze7	67.63	43.49	4.23
syn_haze18	74.56	64.58	8.87

super-pixel in the input image reflects the accuracy of the prediction for the super-pixel. We take the variances of the predicted transmissions for the first GPR layer for analysis and show the histogram distribution of 3315 variances of the four hazy images in Fig 4. The y-axis shows the number of variance values that fall into each interval of the x-axis. Over 70% of variances from the model trained with the T_min set fall into the range between 0 and 0.02, while about 56% from T_mid and 29% from T_max are between the range. Most of the transmission variances by the T_min training set are smaller than the other two training sets. Table 2 shows the mean values of the variances of transmissions for the three input images. We can see that the variances of T_max, T_min, and T_mid are decreasing, which verifies the relationship between the regression accuracy and the similarity of training examples with the input given in Sect. 3.1. The haze removal results of T_max, T_min, and T_mid are shown in Fig 5. We zoom in some details (referring to the red boxes) in the dehazed images. The results from T_mid and T_max either have color distortions or remain a great portion of haze. The dehazed results of T_min have the highest visibility, and the details in the images are restored well.

We also demonstrate quantitative comparisons on different training sets in order to evaluate the applicability of the selection process to the GP regression for dehazing. We calculate the peak signal-to-noise ratio (PSNR) [23] of the dehazed results on the synthetic hazy images to the corresponding original haze-free images in the testing set. We use 27 synthetic hazy images in this experiment, and the box plots of the PSNR values on these images are shown in Fig. 6. The top and bottom lines of the box are the lower and upper quartile values. The horizontal line inside the box indicates the median value while the ends of the whiskers represent the extent of the values. The median values of T_max, T_mid, and T_min are orderly ascending on PSNR values. The T_min set achieves the best restoration, yielding the highest median value among the three. This set learns better mapping from the input to transmission because of the similarity between training examples and the input, indicating the effectiveness of our example searching for input-adaptive dehazing.

5.3 Comparisons with regression using all available images

The results of our work outperform those of [9] that shares a common regression model but trains the model using all available images without any selection. The previous work performs well in some cases, but its accuracy is lower than that using the chosen examples. Some inaccurate estimation of the transmission in [9] may cause the underestimation or overestimation of the transmission, and consequently the dehazed results have haze remained or distortions. In this study, we choose *optimal* training examples for a given input, reducing the variance of the transmission estimation. Figure 7 shows the visual comparisons with the full training set to illustrate the effectiveness of the selection process. In the first and fourth rows of the dehazed results using the full training set, the trees are over-dehazed while the backgrounds are under-dehazed, showing inconsistent quality. The second row of the results using all images has evident color distortions. These unpleasant results can be partially attributed to the inaccurate estimation of the transmission. The use of optimal examples to the input greatly reduces the inaccurate estimation and produces consistent and favorable dehazed results.

Again, we compare the PSNR values of the dehazed results by regressions from the full training set with those from the selected examples. We take the 27 synthetic hazy images in this experiment and demonstrate the box plots of PSNR on these images in Fig. 8. The top lines of the boxes are almost the same, but the bottom line of the full training set is much lower than that of the model from selected training examples. Also, the difference between upper and lower quartiles for the results of full training set is much larger than the difference of those results from selected examples. The lower gap between quartiles indicates the stability of our input-adaptive dehazing with example searching. The selection of training examples ensures the accuracy of transmission estimation,

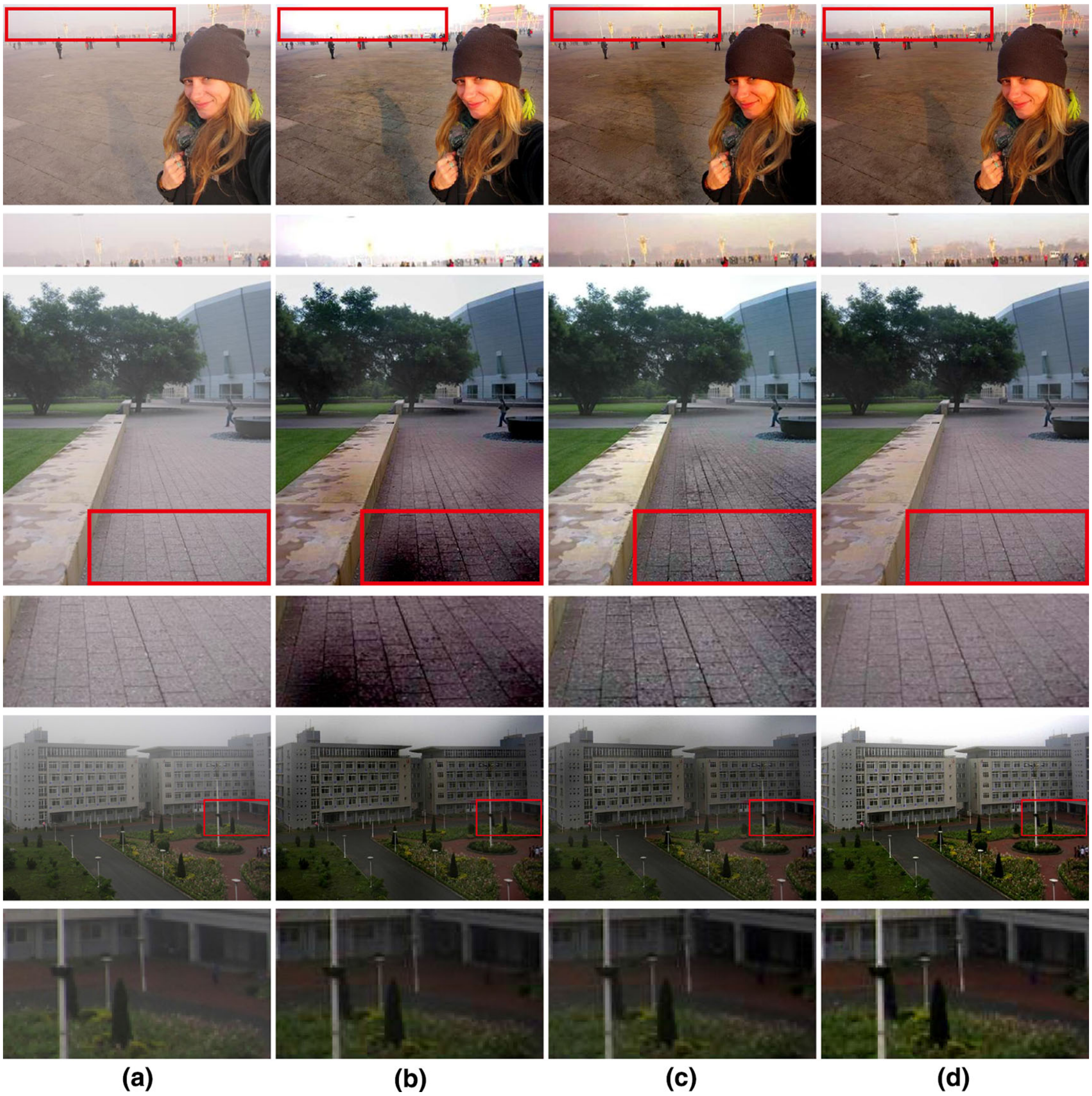

Fig. 5 Visual comparisons on haze removal results of different groups of training examples. The zoomed-in regions (refer to the red rectangles) are illustrated on the second and fifth rows. **a** Hazy images. **b** T_max's results. **c** T_mid's results. **d** T_min's results

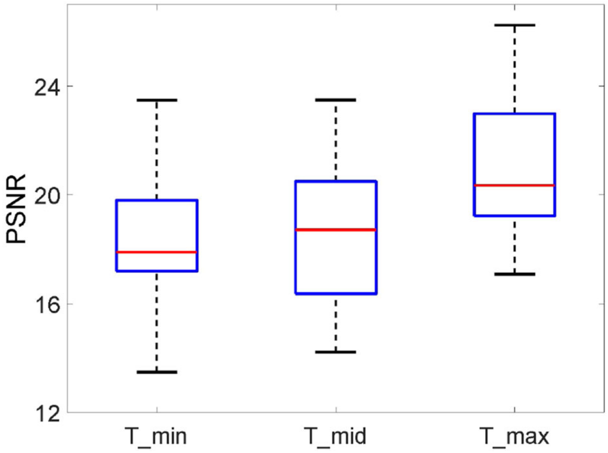

Fig. 6 Box plots of PSNR comparisons between different groups of training examples on synthetic hazy images

and thus the hazy images with various amounts of haze are well restored consistently.

5.4 Comparisons with existing methods

Finally, we compare our input-adaptive haze removal with four latest dehazing methods [22,27,35,39]. Figure 9 shows the resultant images obtained by these different methods. In the first row of Fig. 9, both methods of He and Tang overestimate the thickness of haze and generate dim haze removal results. Those of Zhu and Ren underestimate the transmission, and there exists unpleasant residual haze in the resultant images. In contrast, our dehazing result is quite natural and clear. The regions between the tree and building (referring to the red rectangle in the second row) are severely smeared

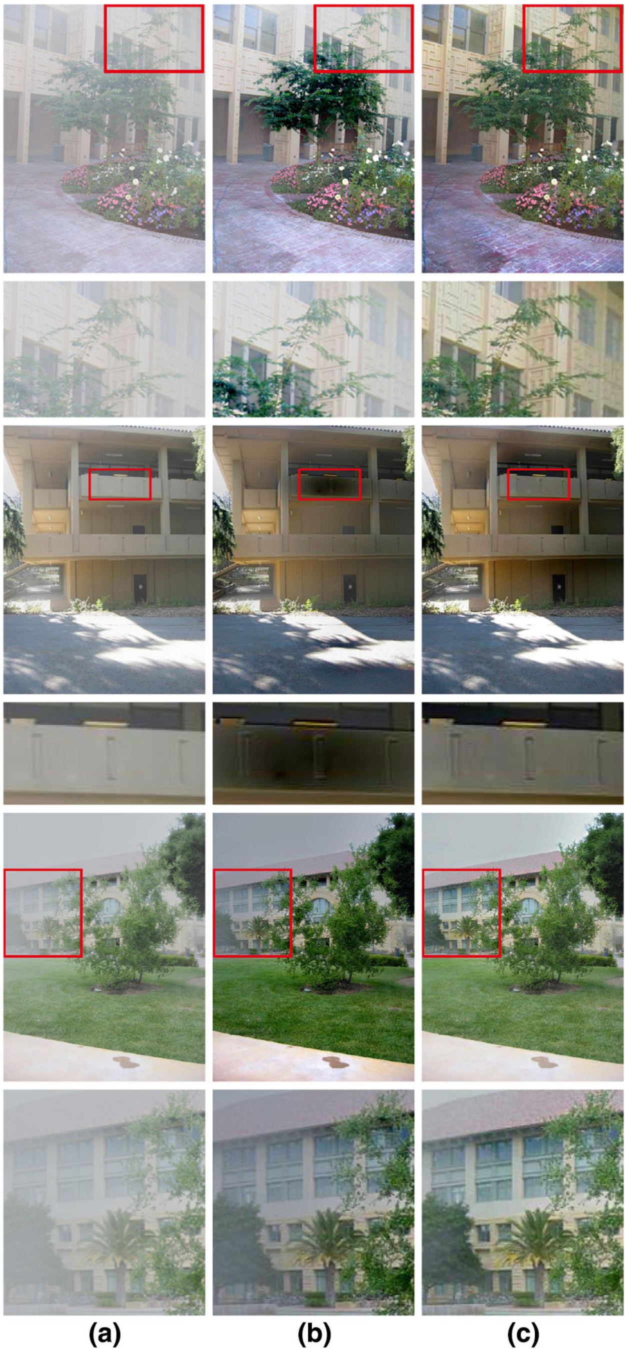

Fig. 7 Visual comparisons with the full training set. The zoomed-in regions (refer to the red rectangles) are illustrated on the third and fifth rows. **a** Hazy images. **b** Full training set. **c** Selected training set

in the other four methods, while our method preserves the details well. In the top-left corner of the image of gym, all the other four methods present a portion of haze effects, but our method restores the region well without any color distortion.

We also exploit the nonreference blur metric [7] to perform the quantitative evaluation on haze removal results of different methods, and the blur metric evaluates the image quality from the perspective of blur perception. When an image is hazy, sharp edges in the image would be smeared out. The blur metric reflects the loss of image details and thus indicates the quality of dehazed images. The lower the value is, the better is the quality of the dehazed image. Actually, nonreference evaluation of dehazing algorithms is still an open issue. There exist several objective metrics as well as subjective rating schemes, but no consensus has yet reached on which one is the best. In our previous work [12], we performed evaluations in terms of two metrices and a subjective survey. These evaluations from different perspectives are largely consistent, especially for regression-based approaches. This paper focuses on the adaptive example selection for regression algorithms. The blur metric, which is easily reproducible, suffices to provide fair evaluations on regression results with and without the example selection.

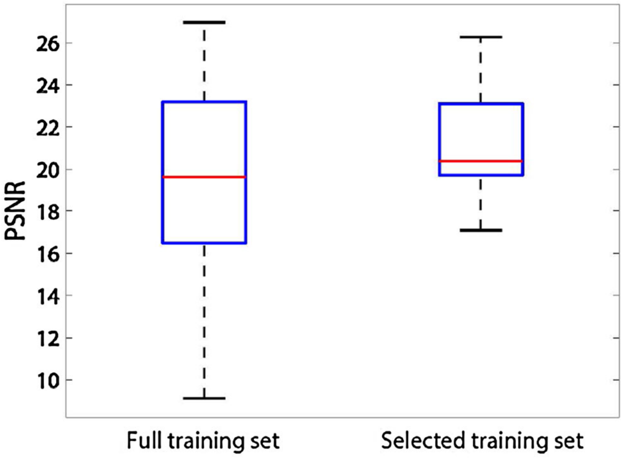

Fig. 8 Box plot of PSNR comparisons with the full training set

We collect 34 real-world hazy images in this experiment for analysis. From the box plots shown in Fig. 10, our results exhibit the lowest median value among all compared methods, showing the effectiveness of the proposed method. Additionally, the proposed method performs quite stable as our dehazed results exhibit the lowest upper and lower quartile values. Similar to [12], we classify the testing images into three categories based on the amount of haze in images and then yield three subsets of testing images: thin, moderate, and dense hazy images. We calculate the mean values of the blur metric on images in the three categories obtained by five dehazing methods as shown in Table 3. Our method has the lowest mean values than the other methods on all three subsets, demonstrating the effectiveness of our method on a wide variety of images with different amount of haze.

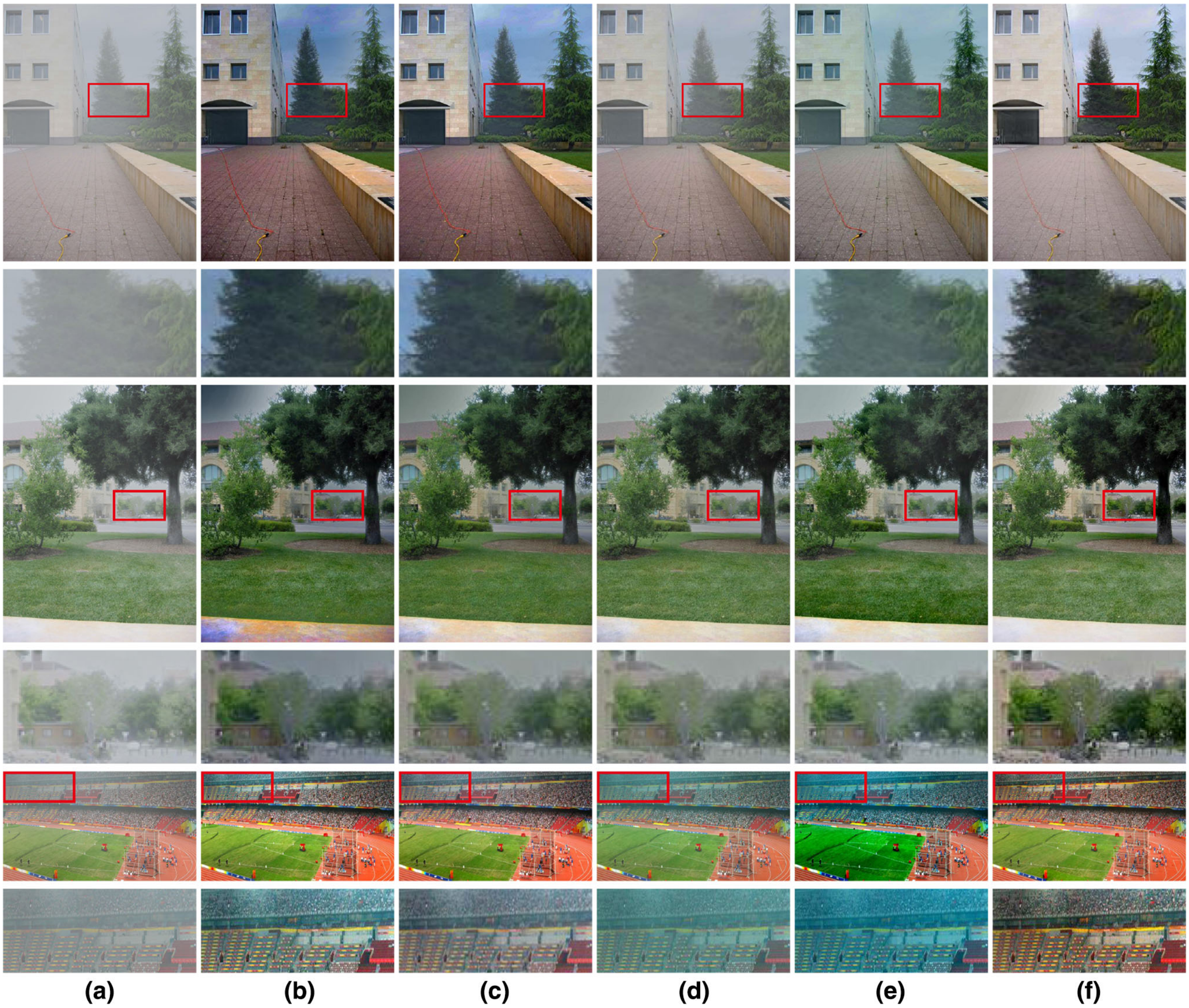

Fig. 9 Visual comparisons on haze removal: **a** input hazy images, **b** He's [22], **c** Tang's [35], **d** Zhu's [39], **e** Ren's [27], and **f** ours. The zoomed-in regions (referring to the red rectangles) are illustrated on the second, fourth, and sixth rows

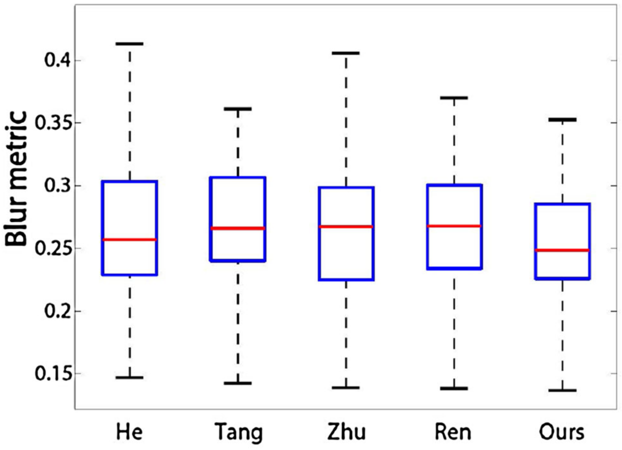

Fig. 10 Box plot of blur metric comparisons on real-world hazy images

Table 3 Mean blur metric of different methods

Test images	He	Tang	Zhu	Ren	Ours
Thin haze	0.2617	0.2697	0.2606	0.2639	0.2511
Moderate haze	0.2737	0.2794	0.2815	0.2808	0.2653
Dense haze	0.2544	0.2694	0.2567	0.2620	0.2501

6 Conclusion

In this paper, we firstly advocate the input-adaptive dehazing that adaptively seeks examples to train a data-driven model specific to a given input and then propose an efficient searching strategy on image examples to learn a more accurate GP regression model for dehazing. The proposed fast searching strategy efficiently finds optimal training examples adaptive to the input. These examples generate GP regressors that predict the target transmission with higher precision, thus yielding improved dehazing performance. The GP model learnt from the chosen examples by the strategy is able to better represent the relationship between the input feature and corresponding transmission, and finally to produce appealing dehazed results. The comparisons with other latest dehazing methods demonstrate the effectiveness of our input-adaptive dehazing with efficient example searching. The idea of searching optimal examples is likely to apply to many data-driven approaches to low-level image processing, where data are always a central issue, in order to improve the performance of respective algorithms.

In the future, we will study optimal example searching algorithms for other regressors of image dehazing. As proved in this paper, using examples close to the input is able to improve the regression accuracy for Gaussian processes so that the optimization of training examples naturally turns out to be the searching of nearest neighbors. For other applications like facial analysis, we designed a sparse model for linear regression [14] and self-reinforced learning strategy for cascaded regression [10]. It is also nontrivial to investigate what are optimal training examples and how to find these examples for regressors other than GP targeting at image dehazing. This investigation directs to our future work.

Acknowledgements This work is partially supported by the Natural Science Foundation of China under Grant Nos. 61572096, 61432003, and 61733002. The authors are grateful to Prof. Ming-Ting Sun at the University of Washington and Dr. Jue Wang at Megvii Inc. for their constructive discussions and suggestions.

References

1. Ancuti, C.O., Ancuti, C., Hermans, C., Bekaert, P.: A fast semi-inverse approach to detect and remove the haze from a single image. In: Asian Conference on Computer Vision, pp. 501–514. Springer (2011)
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Berlin (2006)
3. Burger, H.C., Schuler, C.J., Harmeling, S.: Image denoising: can plain neural networks compete with BM3D? In: Computer Vision and Pattern Recognition, pp. 2392–2399. IEEE (2012)
4. Cai, B., Xu, X., Jia, K., Qing, C., Tao, D.: Dehazenet: an end-to-end system for single image haze removal. arXiv preprint arXiv:1601.07661 (2016)
5. Cao, Y., Brubaker, M.A., Fleet, D.J., Hertzmann, A.: Efficient optimization for sparse Gaussian process regression. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2013)
6. Chen, L., Lu, G., Zhang, D.: Effects of different Gabor filter parameters on image retrieval by texture. In: ACM International Conference Multimedia, pp. 273–278. Citeseer (2004)
7. Crete, F., Dolmiere, T., Ladret, P., Nicolas, M.: The blur effect: perception and estimation with a new no-reference perceptual blur metric. In: Rogowitz, BE., Pappas, TN., Daly, SJ. (eds.) Electronic Imaging 2007, pp. 64,920I–64,920I. International Society for Optics and Photonics (2007)
8. Domingos, P.: A few useful things to know about machine learning. Commun. ACM **55**(10), 78–87 (2012)
9. Fan, X., Gao, R., Wang, Y.: Example-based haze removal with two-layer Gaussian process regressions. In: Keyser, J., Kim, Y.J., Wonka, P. (eds.) Pacific Graphics Short Papers. The Eurographics Association (2014). https://doi.org/10.2312/pgs.20141260
10. Fan, X., Liu, R., Huyan, K., Feng, Y., Luo, Z.: Self-reinforced cascaded regression for face alignment. In: AAAI (2018)
11. Fan, X., Wang, Y., Gao, R., Luo, Z.: Haze editing with natural transformations. Vis. Comput. **32**, 137–147 (2016)
12. Fan, X., Wang, Y., Tang, X., Gao, R., Luo, Z.: Two-layer Gaussian process regression with example selection for image dehazing. IEEE Trans. Circuits Syst. Video Technol. (2016). https://doi.org/10.1109/TCSVT.2016.2592328
13. Fattal, R.: Single image dehazing. ACM Trans. Graph. **27**(3), 72:1–72:9 (2008). https://doi.org/10.1145/1360612.1360671
14. Feng, Y., Liu, R., Fan, X., Huyan, K., Luo, Z.: Leveraging geometric correlation for input-adaptive facial landmark regression. In: IEEE International Conference on Multimedia and Expo (ICME), pp. 385–390 (2017)
15. Freeman, W.T., Jones, T.R., Pasztor, E.C.: Example-based super-resolution. Comput. Graph. Appl. **22**(2), 56–65 (2002)
16. Freeman, W.T., Pasztor, E.C., Carmichael, O.T.: Learning low-level vision. Int. J. Comput. Vis. **40**(1), 25–47 (2000)
17. Frost, R., Armstrong, B.C., Siegelman, N., Christiansen, M.H.: Domain generality versus modality specificity: the paradox of statistical learning. Trends Cogn. Sci. **19**(3), 117–125 (2015)
18. Gibson, K., Belongie, S., Nguyen, T.: Example based depth from fog. In: International Conference on Image Processing, pp. 728–732. IEEE (2013)
19. Gibson, K.B., Belongie, S.J., Nguyen, T.Q.: Example based depth from fog. In: International Conference on Image Processing, pp. 728–732. IEEE (2013)
20. He, H., Siu, W.C.: Single image super-resolution using Gaussian process regression. In: Computer Vision and Pattern Recognition, pp. 449–456 (2011)
21. He, K., Sun, J., Tang, X.: Guided image filtering. In: European Conference on Computer Vision, vol. 35(6), pp. 1397–1409 (2011). https://doi.org/10.1109/TPAMI.2012.213
22. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. IEEE Trans. Pattern Anal. Mach. Intell. **33**(12), 2341–2353 (2012). https://doi.org/10.1109/TPAMI.2010.168
23. Huynh-Thu, Q., Ghanbari, M.: Scope of validity of PSNR in image/video quality assessment. Electron. Lett. **44**(13), 800–801 (2008)
24. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: European Conference on Computer Vision, pp. 304–317. Springer (2008)
25. Kwon, Y., Kim, K.I., Tompkin, J., Kim, J.H., Theobalt, C.: Efficient learning of image super-resolution and compression artifact removal with semi-local Gaussian processes. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1792–1805 (2015)
26. Lazarogredilla, M., Quinonerocandela, J., Rasmussen, C.E., Figueirasvidal, A.R.: Sparse spectrum Gaussian process regression. J. Mach. Learn. Res. **11**, 1865–1881 (2010)
27. Ren, W., Liu, S., Zhang, H., Pan, J., Cao, X., Yang, M.H.: Single image dehazing via multi-scale convolutional neural networks. In: European Conference on Computer Vision (2016)
28. Ren, X., Malik, J.: Learning a classification model for segmentation. In: International Conference on Computer Vision, pp. 10–17. IEEE (2003)
29. Saxena, A., Sun, M., Ng, A.Y.: Learning 3-d scene structure from a single still image. In: International Conference on Computer Vision (2007)

30. Schmidt, U., Rother, C., Nowozin, S., Jancsary, J., Roth, S.: Discriminative non-blind deblurring. In: Computer Vision and Pattern Recognition, pp. 604–611 (2013)
31. Settles, B.: Active Learning Literature Survey. Computer Sciences Technical Report 1648. University of Wisconsin-Madison (2010)
32. Silpaanan, C., Hartley, R.: Optimised kd-trees for fast image descriptor matching. In: Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2008)
33. Sun, L., Hays, J.: Super-resolution from internet-scale scene matching. In: International Conference on Computational Photography, pp. 1–12. IEEE (2012)
34. Tan, R.: Visibility in bad weather from a single image. In: Computer Vision and Pattern Recognition, pp. 1–8 (2008). https://doi.org/10.1109/CVPR.2008.4587643
35. Tang, K., Yang, J., Wang, J.: Investigating haze-relevant features in a learning framework for image dehazing. In: Computer Vision and Pattern Recognition (2014)
36. Tang, X., Fan, X., Duan, Y., Luo, Z.: A fast training example searching algorithm for data-driven dehazing. In: International Conference on Digital Home (2016)
37. Yue, H., Sun, X., Yang, J., Wu, F.: CID: Combined image denoising in spatial and frequency domains using web images. In: Computer Vision and Pattern Recognition, pp. 2933–2940 (2014)
38. Zhao, X., Wang, S., Li, S., Li, J.: Passive image-splicing detection by a 2-d noncausal Markov model. IEEE Trans. Circuits Syst. Video Technol. **25**(2), 185–199 (2015)
39. Zhu, Q., Mai, J., Shao, L.: A fast single image haze removal algorithm using color attenuation prior. IEEE Trans. Image Process. **24**(11), 3522–3533 (2015)
40. Zoran, D., Isola, P., Krishnan, D., Freeman, W.T.: Learning ordinal relationships for mid-level vision. In: International Conference on Computer Vision, pp. 388–396 (2015)

Xin Fan received the B.E. and Ph.D. degrees in Information and Communication Engineering from Xi'an Jiaotong University, Xi'an, China, in 1998 and 2004, respectively. He was with Oklahoma State University, Stillwater, from 2006 to 2007, as a post-doctoral research fellow. In 2008, he was with the Southwestern Medical Center, University of Texas, Dallas, for the second postdoctoral training. He joined the School of Software, Dalian University of China, Dalian, China, in 2009, where he is currently a Full Professor and the Duty Dean of the DUT-RU International School of Information and Science Technology. His current research interests include computational geometry and machine learning, and their applications to low-level image processing and medical image processing.

Xianxuan Tang received the B.E. degree in Software Engineering from Dalian Maritime University, China, in 2014, and M.E. degree in the same major from Dalian University of Technology in 2017. Now she is working at Meituan Inc.

Minjun Hou received the B.E. degree in Software Engineering from Dalian University of Technology, China, in 2017. Now she is studying as a postgraduate student in Dalian University of Technology. Her research interests include machine learning and computer vision.

Zhongxuan Luo received the B.S. degree in Computational Mathematics from Jilin University, China, in 1985, the M.S. degree in Computational Mathematics from Jilin University in 1988, and the Ph.D. degree in Computational Mathematics from Dalian University of Technology (DUT), China, in 1991. He has been a Full Professor of the School of Mathematical Sciences at Dalian University of Technology since 1997. He is currently the Assistant President of DUT. His research interests include computational geometry and computer vision.