

Forecasting Outdoor Scenes with Support Vector Regression and the Radial Basis Function

Chen-Chung Liu

Chin-Yi university of Technology

Associate Professor

ccl@ncut.edu.tw

Kai-Wen Chuang

Chin-Yi university of Technology

Student

n0703124@yahoo.com.tw

Chih-Chin Chang

Chin-Yi university of Technology

Graduate

linked_15@hotmail.com

Chung-Yen Tsai

Chin-Yi university of Technology

Student

kl44kl44@yahoo.com.tw

摘要

支持向量迴歸 (support vector regression) 是由 Vapnik 於 1995 年提出的一種類神經演算法，基於其只使用少量訓練資料即可無限接近零誤差之迴歸特性及其普遍化的特性，而廣泛流行於當今類神經演算法世界中而頗具好評。本研究採取以向量基函數為核的支持向量迴歸演算法，以色彩方塊版在不同時間所攝之影像當作 SVR 神經網路的訓練資料，訓練出一個能由一張風景照及其拍攝時間來預測此風景在任何時間之風景影像。我們首先各將色彩方塊版影像的 R、G、B 成份經離散餘弦轉換 (Forward Discrete Cosine Transform) 後當作 SVR 的訓練資料。以徑向量基底函數作為 SVR 的標準這個類神經網路。訓練完成後的 SVR 網路能夠快速精確的預測各風景照在不同時段的對應 DCT 係數，最後，講這些預測的 DCT 係數做逆向 DCT 轉換，既可模擬出此風景照在各不同時段所呈現的景色。

關鍵詞：離散餘弦變換、支持向量迴歸、徑向量基底函數、類神經網路、風景。

Abstract

In this paper, a novel strategy for forecasting outdoor scenes is introduced. It is an approach combining the support vector regression in neural network computation, the discrete cosine transform. In 1995, Vapnik first introduced a neural-network algorithm called support vector machine (SVM). During recent years, due to SVM's high generalization performance and its attractive modeling features,

it have received increasing attention in the application of regression estimation. The regression estimation of SVM called support vector regression (SVR). A set of color-block images were transformed by the discrete cosine transformation, they are used as the training data for SVR. We also use the radial basis function (RBF) of the training data as the SVR's kernel to establish the RBF neural network. Finally, the time scenes simulation algorithm (TSSA) is able to synthesize the corresponding scene of any assigned time of the original outdoor scene image.

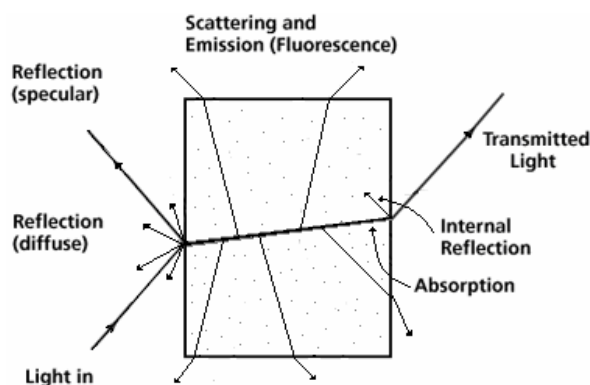
To explore the utility and demonstrate the efficiency of the proposed algorithm, simulations under various input images are conducted. The experiment results show that our proposed algorithm can precisely simulate the desired scenes at an assigned time and has three advantages: (A) Using the color-block images instead of using the scene images of a place to create the reference database, the database can be used for any outdoor scene image taken at anywhere at anytime. (B) Taking the support vector regression on the DCT coefficients of scene images instead of taking the multiple regression on the spatial pixels of scene images, it simplifies the support vector regression procedure and save the processing time.

Keywords: Discrete Cosine Transform, Support Vector Regression, Neural Network, Radial Basis Function (RBF), Scene.

1. Introduction

Sunrise and Sunset are the most beautiful scenes of a day, but they appear only in a very short time interval and people are almost always too busy to wait for their coming. If one can construct the sunset (or sunrise) scene of a scene image taken at anytime, then people shall no longer regret of missing the beautiful moments of their life.

The most special difference between a noon scene and a dusk (or dawn) scene is the colors of the sun and the sky. The Earth's atmosphere contains both air molecules and tiny aerosols (much tinier than the wavelength of the visible light-ray from the Sun), both of which scatter the visible light-ray from the Sun which grazes the Earth's atmosphere; Figure 1 shows the phenomena (scattering, reflection, transmitting and absorption) of a sunlight's beam striking a partially transparent material (the Earth's



atmosphere).

Fig. 1 the phenomena of a sunlight's beam striking a partially transparent material block. (Taken from "The Physics and Chemistry of Color". Kurt Nassau (John Wiley & Sons, New York, 1983).)

The scattering from both air molecules and aerosols is predominantly the Rayleigh scattering [1- 4]. The intensity I of light scattered by a single tiny particle from a beam of light is given by

$$I = I_0 \frac{(1 + \cos^2 \Phi)}{2R^2} \left(\frac{2\pi}{\lambda} \right)^4 \left(\frac{n^2 - 1}{n^2 + 2} \right)^2 \left(\frac{d}{2} \right)^6 \quad (1)$$

where I_0 is the original intensity of entrance light, Φ is the scattering angle between the direction of incidence and the direction of observation, R is the large distance from the particle to the observer, λ is the wavelength of entrance light,

n is the refractive index of particle, and d is the diameter of the particle. The Rayleigh scattering coefficient for a group of scattering particles is the multiplication product of the concentration of particles and the cross-section.

According to the Rayleigh scattering law of light, the scattered light is polarized at a large angle with respect to the direction of the sun's light and blue light is scattered more strongly than red light. When the sun high in the sky, sunlight goes through very little atmosphere, so little scattering takes place, the sky far away to the sun appears light blue and the sky close to the sun appears mostly white (the sun's color). When the Sun is low in the sky at sunrise or at sunset, the Sun's light has to travel through a much thicker layer of atmosphere to reach us on the ground owing to the oblique angle.

This extra distance causes multiple scatterings of blue light, the sky appears less bright and redder than that at noon because more blue light than red light is scattered out of the solar light-rays, leaving an excess of red light.

It is very difficult to obtain the concentration and the diameters of the air molecules and the tiny aerosols in the Earth's atmosphere, so it is impossible to estimate the colors of an outdoor scene precisely by using the Rayleigh scattering law of light. In 2002, Thompson derived a spatial post-processing algorithm of night scene [5] based on the visual noise (VN) and the loss of acuity (LA). There are four strategies in Thompson algorithm: (A) color space transform (CST), (B) day to night conversion (DNC), (C) night filter (NF), and (D) Gaussian noise generator (GNG). It can generate

more realistic night scene, but it neglects the time factor so it cannot clearly illustrate the differences between early night, middle night and midnight. Alexander Toet used the method introduced by Welsh et al to transfer a source image's color characteristics to a grayscale target image (night vision imagery) [6,7]. They search a color source image that resembles the target scenes first. Second, they transformed both the source image and the target image into a perceptually miscorrelated color space. Third, they searched for the best matching source pixel for each target pixel from the source image by using the first order statistics of the luminance distribution in a 7×7 window around the source and target pixels. Finally, they assigned the matching pixel's chromaticity values of the source image to the target pixel. Although they effectively give single-band intensified night vision gray imagery a full color day-time image, but to find a suitable resembled color source image for the target image is not easy, and the final colors of the chromatic target image completely depends on the colors of the source image. They are not the true colors of the target image taken in day. Other papers about the scenes can be found in [8-19].

In 1995, Vapnik first introduced a neural-network algorithm called support vector machine (SVM), which is a novel learning machine based on statistical learning theory [20,21]. The SVM possesses outstanding advantages; (a) the strong theoretical basis provides with high generalization capability and avoids overfitting, (b) the global model can deal with high-dimensional input vectors efficiently, (c) the solution is light and only a subset of training samples contributes to this solution, thus reducing the workload [22].

During recent years, due to SVM's high generalization performance and its attractive

modeling features, SVMs have received increasing attention in areas ranging from its original application of pattern recognition to the extended application of regression estimation[23]. The regression estimation of SVM, called support vector regression (SVR), can only be realized if a suitable kernel function is applied. This kernel function transforms the non-linear input space to a high dimensional feature space in which non-linear relationships can be represented in a linear form. However, there are many kernel functions. The most popular kernel functions are the inner-product based linear and polynomial functions and the Radial Basis Function (RBF) [24].

There are lots of authors used SVR to do the time series forecasting. Zhongsheng et al. apply it for forecasting intermittent demand of spare parts to get accurate forecasts [23] Bao et al. extend the application of SVR for forecasting intermittent demand [25]. Cheng et al. exploit SVR for the time series forecasting to the processing of end effects of Hilbert-Huang transform to obtain an excellent effect. [26]. Pai and Lin used the SVR model in forecasting stock prices problems [27]. And, there still has many applications of SVR on other areas [28- 32].

The red, green and blue are the primary components of light (or color image) in spatial domain. It is reasonable to take the support vector regression on each component individually. But, the spatial pixels of an image are often correlative, so the spatial domain of an image is not suitable for taking support vector regression. The DCT has properties of noise filtering, miscorrelation, and feature preservation. It is often used as a preprocessing step followed by a sophisticated algorithm in image processions, and it always has good performance while one combines the DCT block transformation with other techniques. In some cases, it has been

showing that the processing based on DCT has significantly better results than the directive pixel based processing [33,34]. To take time scene simulation on the DCT domain by using support vector regression will be a good idea. The remainder of this paper is organized as follows: Section 2 presents the the time scenes simulation algorithm (TSSA). Empirical results is described in the Section 3. Finally, Section 4 concludes this paper.

2. The Algorithm of Time Scenes Simulation (TSSA)

2.1 The RGB color system

The basic understanding of color perception of the human visual system shows the fact, which human eyes pick out the color of an object illuminated by white light is the reflection of selected wavelengths of light by that object [35-36]. The appearance color of an object can be considered as the object absorbing all colors except the colors that are reflected. For example, a red object illuminated by white light absorbs nearly all of the wavelengths except those corresponding to red light. Although there are an unlimited number of colors just as there are immeasurable quantities of real numbers on a number line, colors of the range of human visual sensations can be produced by the mixtures of finite visual lights of various wavelengths. Human eyes have three cones shaped light detectors [37]. Each cone is sensitive to a special range of colors. One is sensitive to primarily red, a second to primarily green, and a third to primarily blue. The three sensitivity ranges overlap in fact. Each cone transmits a nerve pulse to the brain at a rate proportional to the light's intensity of that cone detects. By combining the signal rates transmitted from three cones, the brain attempts to conclude what color and brightness of the light must be. Thus, to combine

various quantities of three colors (red, green and blue) of light can turn out all colors that can be perceived for human.

In televisions, computer monitors, and colored image projection systems, only the three colors are enough to adequately represent any of the unlimited visible colors [38]. For measuring or reproducing color, a number of three dimensional color models have been defined; Color models like as HIS, $L^*a^*b^*$, YIQ, YUV, and YCbCr are suitable for image processing applications, they are the reducing redundancy models of the image in RGB color space, obtained by some color transform from RGB color space, They are suitable for image processing applications, but not suitable for the color representation of a color image. The image in RGB color space is not suitable for image processing applications, because the image in RGB color space is highly correlated, but it is the most suitable for the color representation of a color image. Most imaging devices (CCD cameras, CRT monitors, etc) mimic the retina and utilize three separate channels calibrated for the detection of red, green and blue phosphors to render colors at every pixel on the screen [39]. The RGB color model is the most popular and natural color model. Due to it can compose any color adequately; it is frequently used in color image representation. R, G and B component of a color are given by:

$$R, G, B = k \int_{400}^{700} I(\lambda) \Phi(\lambda) S_{R,G,B}(\lambda) d\lambda \quad (2)$$

Where k is a constant that defines the total overall brightness response of the human eyes, $I[\lambda]$ is the illumination spectral intensity of a color, $\Phi[\lambda]$ is the object spectral reflectivity, $S_{R,G,B}[\lambda]$ is the spectral sensitivity of the R or G or B channel of the detector and λ is the wavelength. The normalized red, green, and blue coordinates are defined as follow:

$$r = R/(R + G + B) \quad (3)$$

$$g = G/(R + G + B) \quad (4)$$

$$b = B/(R + G + B) \quad (3)$$

where R, G and B are the intensities of red, green, and blue light at a given pixel [40]. We demonstrated that color images could be represented using three primary colors.

In order to obtain the more accuracy regression of a scene image, the original image is

first spilt into its R, G, and B components. Its final simulation image is reconstructed by composing these regression color components.

2.2 The support vector regression

The SVR is an adaptation of support vector machines (SVM). The structure of the SVR is showed in figure 2.

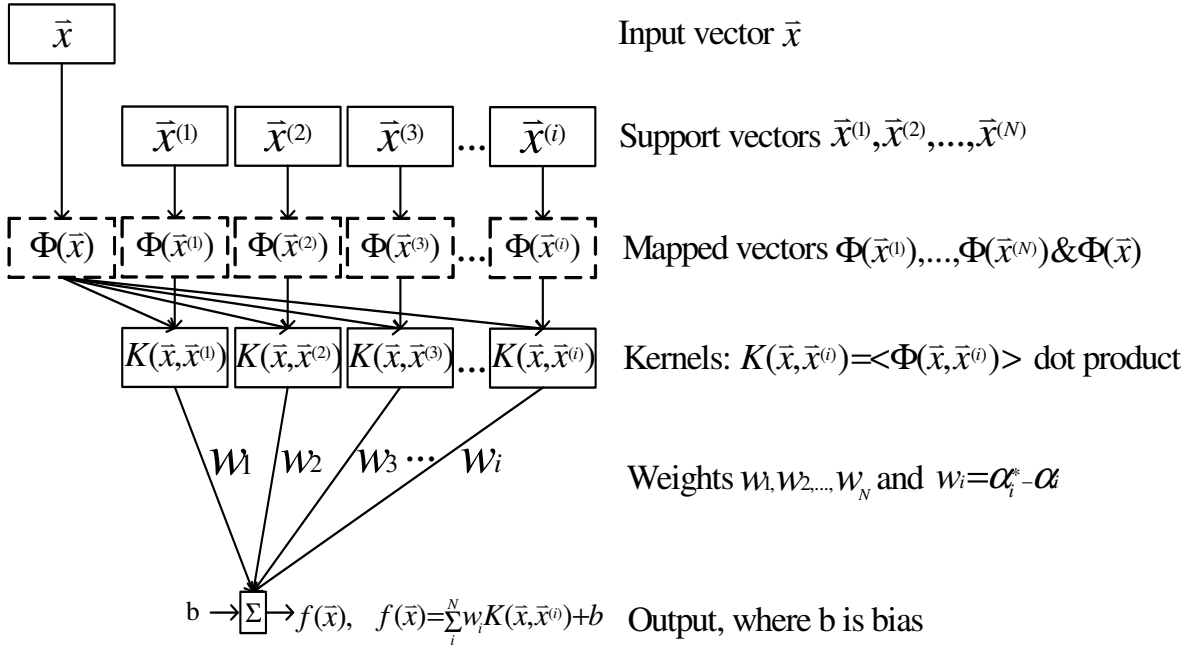


Fig. 2 The structure of support vector machine for regression

Consider a training data set $T = \{(\bar{x}^{(i)}, t_i)\}_{i=1}^N$, $\bar{x}^{(i)} \in R^n, t_i \in R$, where $\bar{x}^{(i)}$ is a vector of input variables and t_i is the corresponding scalar output (target) value. The objective over here is to construct a SVR model to fit a regression function, $t = f(\bar{x})$, such that it can accurately predict the outputs, $\{t_i\}_{i=1}^N$ corresponding to the input vectors $\{\bar{x}^{(i)}\}_{i=1}^N$. With this objective, the linear SVR formula can be given as

$$f(\bar{x}) = w \cdot \Phi(\bar{x}) + b \quad (6)$$

where f is the regression function to be built, w the weight vector in feature space, Φ is the transformation function that transfer input vectors into the high dimension feature space, $w \cdot \Phi(\bar{x})$ is the inner product of w and $\Phi(\bar{x})$, and b is

the bias (constant). The steps to establish the regression is as follow:

Step 1: Choose proper tolerance parameter ε , disciplinary parameter C , and loss function $l(\bar{x}, t, f)$. Then, define empirical risk R_{emp} as

$$R_{emp} = \sum_{i=1}^N l(f(\bar{x}^{(i)}) - t_i) \quad (7)$$

The most popular loss function is the linear ε -insensitive loss function, it is expressed as

$$l(\bar{x}, t, f) = |t - f(\bar{x})|_{\varepsilon} = \begin{cases} 0, & |f(\bar{x}) - t| \leq \varepsilon \\ |f(\bar{x}) - t| - \varepsilon, & |f(\bar{x}) - t| > \varepsilon \end{cases} \quad (8)$$

Step 2: Choose suitable kernel function $k(\bar{x}^{(i)}, \bar{x}^{(j)}) = \Phi(\bar{x}^{(i)}) \cdot \Phi(\bar{x}^{(j)})$. There are several kernel functions including polynomial kernel, radial basis function and sigmoid kernel, etc.[41] The most common kernel is the radial basis function, its mathematical expression is

$$k(\bar{x}^{(i)}, \bar{x}^{(j)}) = \exp(-\|\bar{x}^{(i)} - \bar{x}^{(j)}\|^2 / 2\sigma^2) \quad (9)$$

Step 3: Minimize the risk function

$$R(w) = R_{emp} + \frac{1}{2} \|w\|^2 = \sum_{i=1}^N l(f(\bar{x}^{(i)}) - t_i) + \frac{1}{2} \|w\|^2 \quad (10)$$

After some reformulations and introduction of Lagrange multipliers satisfying the constraints $\sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0$ and $0 \leq \alpha_i$, $\alpha_i^* \leq \frac{C}{N}$, $i=1,2,\dots,N$. Then, the dual problem is resolved, that is:

Minimize:

$$Q(\alpha, \alpha^*) = \frac{1}{2} \sum_{i=1}^N (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) k(\bar{x}^{(i)}, \bar{x}^{(j)}) + \varepsilon \sum_{i=1}^N (\alpha_i^* + \alpha_i) - \sum_{i=1}^N t_i (\alpha_i^* - \alpha_i) \quad (11)$$

Subject to $\sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0$ and $0 \leq \alpha_i$,

$$\alpha_i^* \leq \frac{C}{N}, \quad i=1,2,\dots,N \quad (12)$$

Step 4: Establish the SVR model

$$f(\bar{x}) = \sum_{i=1}^N (\bar{\alpha}_i^* - \bar{\alpha}_i) k(\bar{x}^{(i)}, \bar{x}^{(j)}) + \bar{b} \quad (13)$$

Where

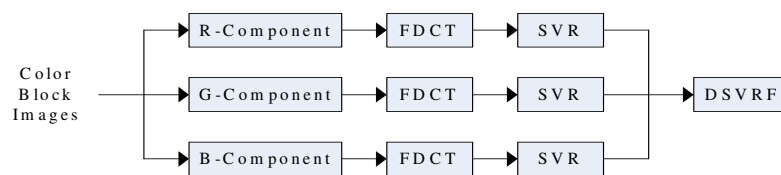
$$\bar{b} = t_j - \sum_{i=1}^N (\bar{\alpha}_i^* - \bar{\alpha}_i) (\bar{x}^{(i)} \cdot \bar{x}^{(j)}) + \varepsilon, \text{ for } \bar{\alpha}_i \in (0, C/N) \quad (14)$$

$$\bar{b} = t_k - \sum_{i=1}^N (\bar{\alpha}_i^* - \bar{\alpha}_i) (\bar{x}^{(i)} \cdot \bar{x}^{(k)}) - \varepsilon, \text{ for } \bar{\alpha}_i^* \in (0, C/N) \quad (15)$$

2.3 The construction of support vector regression function database

This paper proposed a way to construct the sunset scene of a scene image taken at anytime by TSSA. Fig.3 is the flow chart of the construction of the support vector regression functions database. At the beginning of the process, the R, G, and B planes of the

decomposed color input image were split into non-overlapping 8×8 blocks, and then transferred into the frequency domain from the spatial domain by block DCT respectively. In the DCT domain, most of the signal information tends to be concentrated in a few low-frequency components. And the DC coefficient carry information about average luminance within the DCT blocks, the DC coefficient becomes the most meaningful when considering how the luminance is changing between the images [33, 34]. We took the average of the DC components of all the 8×8 DCT blocks to serve as the average value of the reflective color at that time. Then, we used support vector regression to find support vector regression functions of the DCT coefficients of these color-block images that were regularly photographed by fixed time interval from 17:00 to 19:30 of a day. These support vector regression functions were then constructed into a table as the database. Fig.4 shows the image of a color-block taken at 17:20, its R, G, B component images, and the images of DCT coefficients of the R, G, B component images. Fig.5 is the figure of support vector regression functions of the DCT coefficients. Table 1-3 are the tables of the average magnitudes of the 8×8 DCT coefficients for the R, G, B component images of the color-block. There are 64 bases for each table, the upper left base is the block's mean and it is also called the DC value of the image, another 63 bases are called the AC values of the image. Table 1-3 show that the DC value is much larger than the AC values.



FDCT : Forward Discrete Cosine Transform SVR : Support Vector Regression
DSVRF : Database of Support Vector Regression Function

Fig. 3 The construction of the support vector regression functions database

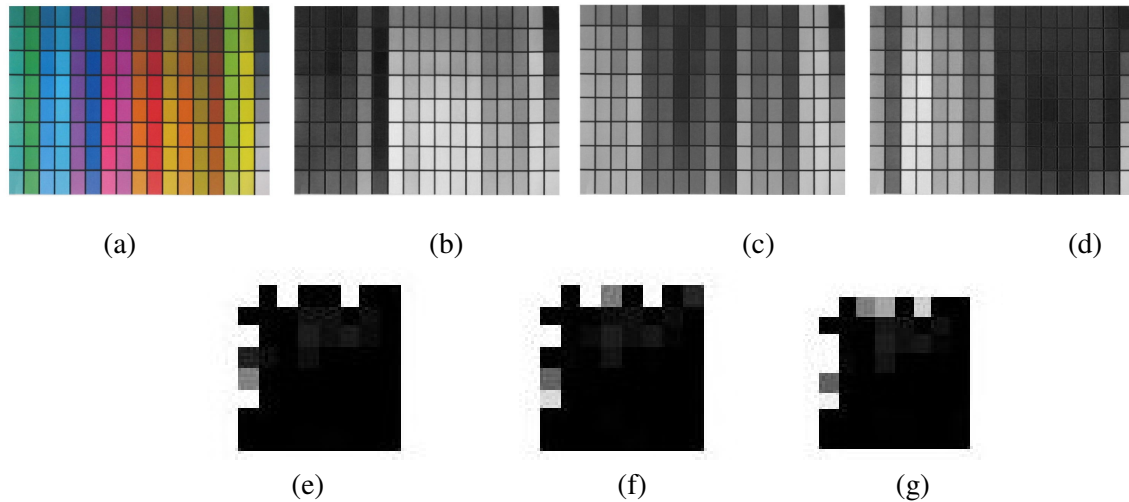


Fig. 4 The image of a color- block taken at 17:20(a) real image (b) R component image(c) G component image (d) B component image(e)The image of DCT coefficients of the R component(f)The image of DCT coefficients of the G component(g)The image of DCT coefficients of the B component.

Table 1 The DCT coefficients for the R component of the color-block image in Fig.4.

945.6788	-3.11356	2.458706	-0.33716	-1.17563	1.24538	-0.23217	-0.927
-5.97103	-0.1986	-0.02393	0.069363	0.06657	-0.00409	0.057666	-0.00114
16.23694	-0.16898	-0.09017	0.10713	0.059559	0.099738	0.041851	-0.04458
0.121902	0.048323	-0.05908	0.059259	-0.03403	0.005014	0.003166	-0.01487
0.504839	-0.04899	0.000112	-0.01591	-0.01855	0.003226	-0.00636	0.00413
1.000081	-0.02425	0.00065	-0.01323	0.0005	-0.00932	0.000607	0.003872
-0.23579	-0.00222	-0.02985	-0.00037	0.003775	0.001865	0.000458	0.003585
-0.48485	-0.01236	-0.00196	0.004875	0.008721	-0.0107	-0.00418	0.006822

Table 2 The DCT coefficients for the G component of the color-block image in Fig.4.

857.0752	-1.16039	2.932919	0.480016	-1.19657	1.780491	-0.24423	0.147122
-5.24403	-0.09254	-0.00674	0.0505	0.059852	-0.00823	0.049707	-0.01809
13.04633	-0.1286	0.039403	0.10974	0.055904	0.090136	0.021017	-0.02801
-0.13035	0.011812	-0.05759	0.071962	-0.03963	0.006384	0.003408	-0.01158
0.38559	-0.03216	0.005886	-0.01135	-0.01836	0.004468	0.000146	0.00476
0.849799	-0.02725	-0.00554	-0.01184	-0.00021	-0.01611	-0.00312	-0.00392
-0.27928	-0.01216	-0.03458	0.009935	0.002494	-0.01045	0.006994	-0.00207
-0.51225	-0.00888	-0.00602	-0.0013	0.007954	-0.00337	0.001674	0.006048

Table 3 The DCT coefficients for the B component of the color-block image in Fig.4.

756.0942	-1.61977	0.49209	0.649581	-1.17632	0.806817	-0.05249	-1.09367
-9.92855	-0.18361	-0.04784	0.088409	0.064015	0.004417	0.064835	-0.0194
7.803619	-0.11323	-0.01613	0.111271	0.064356	0.089781	0.027291	-0.05435
2.570166	0.024903	-0.05725	0.078386	-0.03275	-0.00292	0.0057	-0.01229
0.369813	-0.0401	-0.00135	0.000356	-0.01725	0.002273	-0.00308	0.003574
0.945137	-0.02902	-0.00551	-0.00782	0.000382	-0.0164	0.000275	-6.6E-05
-0.23514	-0.01996	-0.03039	0.009302	0.005829	-0.00298	0.00189	0.010154
-0.48137	-0.01432	-0.00211	0.005969	0.005762	-0.00335	-0.00572	0.005044

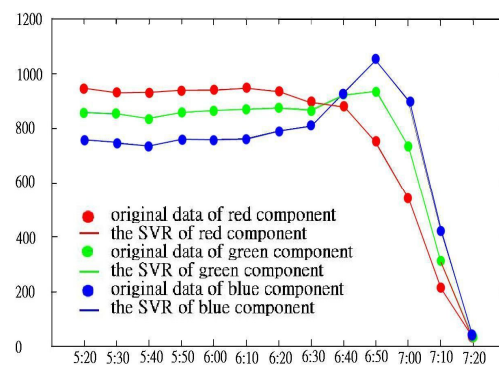
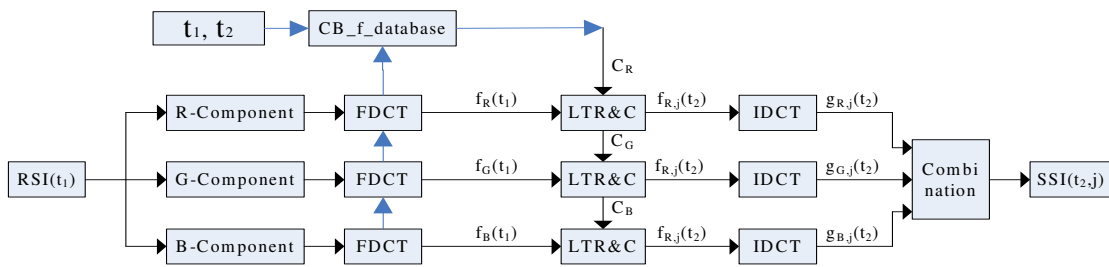


Fig. 5 The figure of support vector regression functions of the DCT coefficients, * indicate the DCT coefficients of these color-block images.

2.4 The simulation image synthesis of time scene

Fig. 6 is the flowchart of the simulation image of time scene. To synthesize the scenes at specific time from the scene taken at any daytime, the input image was divided into R, G, and B planes firstly. The R, G, and B components of a scene photographed at time t_1 are processed respectively with the same steps; they were transferred to the frequency domain by forward discrete cosine transform (FDCT) firstly. Then compared the DC component with the DC

component of color-block at time t_1 to find the compensated coefficient C_1 , to find the corresponding DC component $f_{i,j}(t_2)$ at time t_2 from looking up the database table, and then to combine $f_{i,j}(t_2)$ and C_1 to determine the final R, G, B components $g_{i,j}(t_2)$ of the simulation scene at time t_2 on frequency domain. Finally, we took the inverse discrete cosine transform (IDCT) on $g_{i,j}(t_2)$, and combined them to form the color simulation image at time t_2 of the scene.



RSI(t_1) : Real Scene Image photographed at time t_1
 $f_R(t_1)$: R-component on frequency domain of image $f(t_1)$
 $f_G(t_1)$: G-component on frequency domain of image $f(t_1)$
 $f_B(t_1)$: B-component on frequency domain of image $f(t_1)$
 $f_{R,j}(t_2)$: The j-th degree simulation of R component at time t_2 on frequency domain
 $f_{G,j}(t_2)$: The j-th degree simulation of G component at time t_2 on frequency domain
 $f_{B,j}(t_2)$: The j-th degree simulation of B component at time t_2 on frequency domain
IDCT : Inverse Discrete Cosine Transform
 $g_{R,j}(t_2)$: The j-th degree simulation result of R component at time t_2 on R.G.B domain
 $g_{G,j}(t_2)$: The j-th degree simulation result of G component at time t_2 on R.G.B domain
 $g_{B,j}(t_2)$: The j-th degree simulation result of B component at time t_2 on R.G.B domain
SSI(t_2,j) : The j-th degree scene simulation Image at time t_2
CB_f_database : database of Color Blocks on frequency domain
 t_1 : The time of the input image was taken

Fig. 6 The flow chart of the simulation image synthesis of a time scene.

3. Experiment results

Several test images with size (720×512) were used in simulation for demonstrating the performance of the proposed scheme. In order to give a detailed description of the simulation results, Fig. 7 is used to express the simulation results; comparing the corresponding simulation image with the real image, we found some lamplights in the real image taken at 19:00, but not in the corresponding synthesis image. Apart from the lamplights, the human eyes can not feel any difference between them. Table 4 shows the absolute error ratios of simulation image with respect to the real image of Fig.7. From the error table we can find that the error in R plane is the

largest and the error in B plane is the smallest, this is caused by the fact that the average pixel value in B plane is larger than the average pixel value in R plane. On the other hand, the error in order 2 and order 3 best fitting regressions are much larger than the error in order larger than 3 best fitting regressions. This coincides with the Rayleigh scattering theory of sunlight. Experimental results show that there is no difference in visual quality between the simulation image obtained by our algorithm and the real image. However, because our algorithm is carried out in the DCT domain, it is time-saving and simpler.

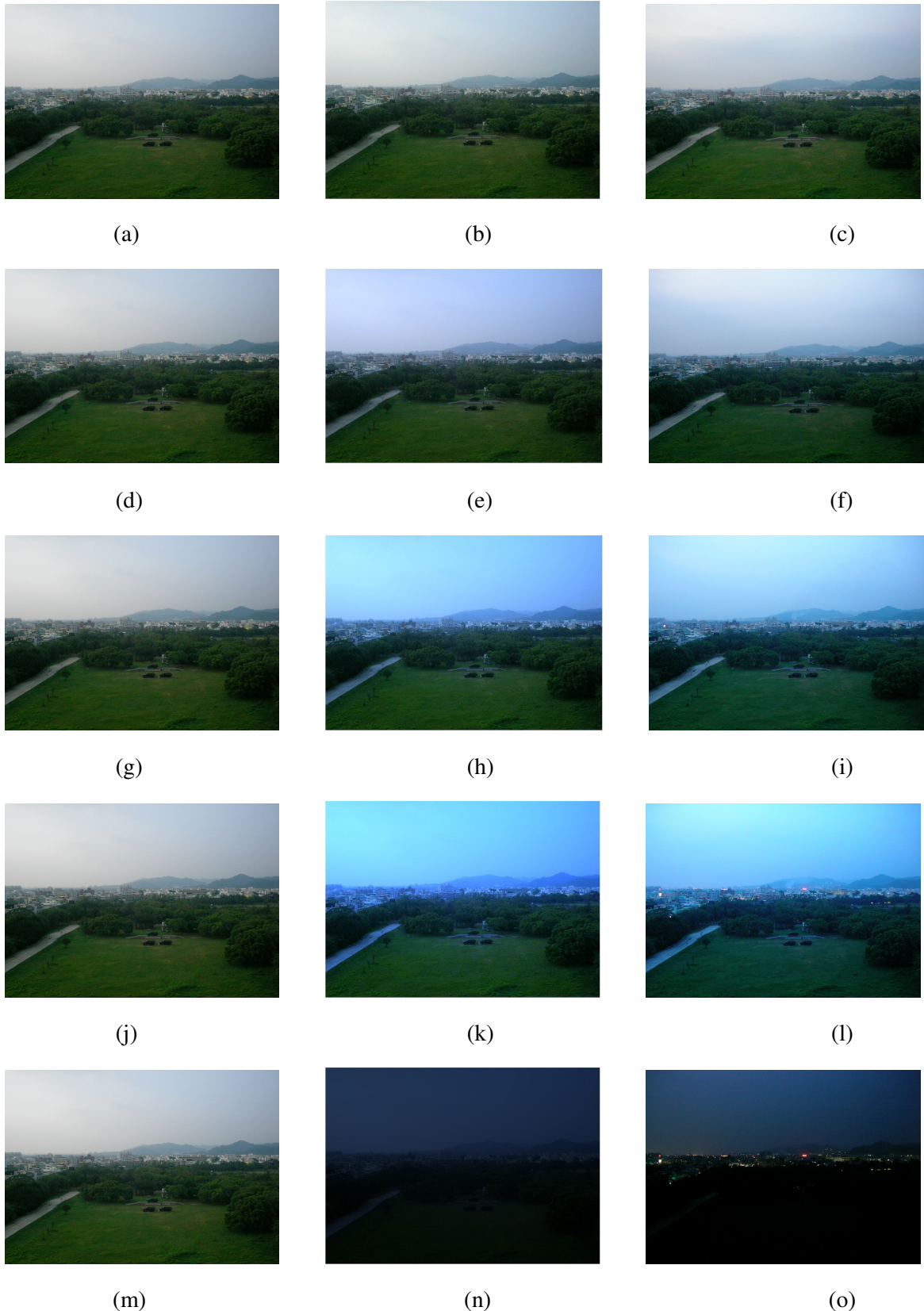


Fig. 7 simulation images, (a), (d), (g), (j), (m) is a scene photographed at 17:30, (b) the simulation image of (a) at 18:00, (c) is the scene photographed at 18:00 of (a). (e) the simulation image of (d) at 18:30, (f) is the scene photographed at 18:30 of (d). (h) the simulation image of (g) at 18:50, (i) is the scene photographed at 18:50 of (g). (k) the simulation image of (j) at 19:00, (l) is the scene photographed at 19:00 of (j). (n) the simulation image of (m) at 19:10, (o) is the scene photographed at 19:10 of (m).

Table 4 Table of absolute error ratios of simulation image with respect to real image of Fig.7.

	18:00	18:30	18:50	19:00	19:10
Error in R plane(%)	10.7404	17.1363	15.4677	17.8973	36.4459
Error in G plane(%)	7.6832	11.8439	6.8233	8.5632	42.5396
Error in B plane(%)	8.2303	9.1075	6.5911	6.6485	27.9848

4. Conclusion

For SVR simulation using a computer with Intel Pentium D 2.80GHz CPU and 512MB RAM, it only takes less than 1 second to simulate a scene image of size 720×512 by our look-up table algorithm. On the other hand, simulating the same image by support vector regression with the same computer takes more than 5 seconds, simulating the same image by multiple regression with the same computer takes more than 540 seconds. Table 4 is the table of absolute error ratios of the SVR simulation with respect to real image. It shows that the absolute error ratio of each degree simulations are always less than 18% with the time distance less than 90 minutes ; this error ratio can not be found by human eyes and it can be reduced by adopting more frequency bands in DCT domain of images for regression and simulation. However, adopting more frequency bands is going to take more time and need more computer memory. With the above experimental results and performance analyses, we conclude that the proposed approach has three major advantages: (A) Using the color-block images instead of using the scene images of a place to create the reference database, this approach is suitable for any outdoor scene taken at anywhere at anytime. (B) Taking the support vector regression on the DCT coefficients of scene images instead of taking the support vector regression on the spatial pixels of scene images, this approach simplifies the support vector regression procedure and saves the processing

time. (C) Using look-up table algorithm to save time. With this algorithm, one can successfully and effectively simulate the anytime scene from a outdoor scene taken at anytime no matter it is of uniform color or complex colors, so it is suitable to be used in the design of virtual reality and the recognition of natural scenes.

References

- [1] Van de Hulst HC., "Light Scattering by small particles," Dover Publications, New York, 1981
- [2] Maarten Snee and Wim Ubachs, "Direct measurement of the Rayleigh scattering cross section in various gases," Journal of Quantitative Spectroscopy & Radiative Transfer, Vol. 92, pp. 293- 310, 2005.
- [3] R. V. Deelen, J. Landgraf and I. Aben, "Multiple elastic light scattering in the Earth's atmosphere: a doubling- adding method to include rotational Raman scattering by air," Journal of Quantitative Spectroscopy & Radiative Transfer, Vol. 95, pp. 309- 330, 2005.
- [4] S. M. Lea and J. R. Burke, "Physics: The nature of things", annotated instructor's edition 1997, West company, USA, pp. 597-626.
- [5] W. B. Thompson, P. Shirley, and J. A. Ferwer, "A Spatial Post- Processing Algorithm for Images of Night Scenes," Journal of Graphics Tools, Vol. 7, No. 1, pp. 1- 12,

- 2003.
- [6] T. Welsh, M. Ashikhmin, K. Mueller, Transferring color to greyscale images. Proceedings of 29th annual conference on computer graphics and interactive techniques, ACM Press, New York, USA, 2002, pp. 277-280.
- [7] Alexander Toet, 'Colorizing single band intensified nightvision images', Displays, Vol. 26, No. 1, pp: 15-21, 2005
- [8] Arthur R. Weeks, Jr., "Fundamentals of Electronic Image Processing", SPIE/IEEE Series on Image Science & Engineering, Bellingham, Washington USA, pp. 440-451, 1998.
- [9] Angus M. Brown, "A step-by-step guide to non-linear regression analysis of experimental data using a Microsoft Excel spread sheet", Computer methods and programs in Biomedicine 65, pp. 191-200, 2001.
- [10] C. J. Vianna, E. A. De Souza, Combining Marr and Canny Operators to Detect Edges, Microwave and Optoelectronics Conference, 2003. Vol. 2, pp. 695-699.
- [11] E. Bourennane, P. Gouton, M. Paindavoine, F. Truchetet, Generalization of Canny-Deriche filter for detection of noisy exponential edge, Signal processing 82 (2002) 1317- 1328.
- [12] G. Qiu and J. Duan, "An Optimal Tone Reproduction Curve Operator for the Display of High Dynamic Range Images," IEEE International Symposium on Circuits and Systems, pp.6276-6279, 2005.
- [13] G. W. Larson, H. Rushmeier, and C. Piatko, "A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes," IEEE Trans. On Visualization and Computer Graphics, Vol. 3, No. 4, pp.291-236, 1997.
- [14] H. Zhang and L. Lucchese, "A Fast Tone Reproduction Algorithm for High Dynamic Range Image Display," 2004 IEEE 6th Workshop on Multimedia Signal processing, pp. 275-278. 2004.
- [15] J. A. Ferwerda, "Elements of Early Vision for Computer Graphics," IEEE Computer Graphics and Applications, Vol. 21, No.5, pp. 22-33, 2001.
- [16] J. Tumblin and H. Rushmeler, "Tone Reproduction for Realistic Images, IEEE Computer Graphics and Applications, Vol. 13, No.6, pp. 42-48, 1993.
- [17] M. Boutell and J. Luo, "Sunset Scene Classification Using Simulated Image Recomposition," Proceeding, IEEE Int'l Conf. on Multimedia and Expo, pp.I.37-I.40, 2003.
- [18] M. Hermon, "Generating Detailed Scene Descriptions," IEEE Int'l Conf. On Robotics and Automation, Vol. 2, pp.426-431, 1986.
- [19] R. C. Gonzalez and R. E. Woods, "Digital image processing," 2nd ed., New Jersey: Prentice-Hall, 2002, pp. 675-683.
- [20] Vapnik VN, "The nature of statistical learning theory", New York: Springer, 1995.
- [21] K. Y. Chen, "Forecasting systems reliability based on support vector regression with genetic algorithms", Reliability Engineering and System Safety, Vol. 92, pp. 423-432, 2007.
- [22] T. Zou, Y. Dou, M. Hong, J. Zou, Y. Ren, "Support vector regression for determination of component of compound oxytetracycline powder on near-infrared spectroscopy", Analytical Biochemistry, Vol. 355, pp. 1-7, 2006.
- [23] Z. Hua, B. Zhang, "A hybrid support vector machines and logistic regression approach for forecasting intermittent demand of spare parts", Applied Mathematics and Computation, Vol. 181, pp. 1035-1048, 2006.

- [24] U. Stu, W.J. Melssen, L.M.C. Buydens, "Facilitating the application of Support Vector Regression by using a universal Pearson VII function based kernel B", *Chemometrics and Intelligent Laboratory Systems*, Vol. 81, pp. 29-40, 2006.
- [25] Y. Bao, W. Wang, J. Zhang, "Forecasting intermittent demand by SVMs regression", *IEEE International Conference on Systems, Man and Cybernetics*, pp. 461-465, 2004.
- [26] D. Yu, Y. Yang, "Application of support vector regression machines to the processing of end effects of Hilbert-Huang transform", *Mechanical Systems and Signal Processing*, Vol. 21, pp. 1197-1211, 2007.
- [27] P.F. Pai, C.S. Lin, "A hybrid ARIMA and support vector machines model in stock price forecasting", *Omega*, Vol. 33, pp. 497-505, 2005.
- [28] X. C. Xi, A. N. Poo, S. K. Chou, "Support vector regression model predictive control on a HVAC plant", *Control Engineering Practice*, Vol. 15, pp. 897-908, 2007.
- [29] V.N. Vapnik, S.E. Golowich, A.J. Smola, "Support vector method for function approximation, regression estimation, and signal processing", *Advances in Neural Information Processing Systems*, Vol. 9, pp. 281-287, 1996.
- [30] F.E.H. Tay, L.J. Cao, "Application of support vector machines in financial time series forecasting", *Omega*, Vol. 29, No.4, pp. 309-317, 2001.
- [31] L.J. Cao, F.E.H. Tay, "Support vector machine with adaptive parameters in financial time series forecasting", *IEEE Transactions on Neural Networks*, Vol.14, No.6, pp. 1506-1518, 2003.
- [32] K. Kim, "Financial time series forecasting using support vector machines", *Neurocomputing*, Vol. 55, pp. 307-319, 2003.
- [33] D. Zhong, I. Defée "DCT histogram optimization for image database retrieval", *Pattern Recognition Letters*, Vol. 26, No. 14, pp. 2272-2281, 2005.
- [34] J. Tang, "A contrast based image fusion technique in the DCT domain", *Digital Signal Processing*, Vol. 14, pp.218-226, 2004.
- [35] S. S. Deeb, "Genetics of variation in human color vision and the retinal cone mosaic", *Current Opinion in Genetics & Development*, Vol: 16, No.3, pp. 301-307, 2006.
- [36] Lennie, Peter, "Color vision: Putting it together", *Current Biology*, Vol. 10, No.16, pp. R589-R591, 2000.
- [37] S. D.Buluswar, B. A. Draper, "Color Models for Outdoor Machine Vision," *Computer Vision and Image Understanding*, Vol.85, No.2, pp. 71-99, 2002.
- [38] C. Clausen and H. Wechsler, "Color Image Compression Using PCA and Backpropagation Learning, " *pattern recognition*, Vol. 33, pp1555-1560, 2000.
- [39] T. Bose, "Digital signal and image processing", 2004, J. Wiley & sons, New York
- [40] I. Andreadis and Ph. Tsalides, "Analog computation of Image Chromaticity," *Real-Time Imaging* 3, pp.1-6, 1997.
- [41] A. B.Gandhi, J. B. Joshi, V. K. Jayaraman, B. D. Kulkarni, "Development of support vector regression (SVR)-based correlation for prediction of overall gas hold-up in bubble column reactors for various gas-liquid systems", *Chemical Engineering Science* Vol. 62, No.24, pp. 7078-7089, 2007.