Creating Walk-though Images from Traffic Video Sequence

Yaochen Li, Yuehu Liu, Nanning Zheng

Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, China

Email: liyaochensmile@gmail.com

Abstract—In this paper, we present a novel scheme which can let users experience the feel of navigating into traffic scenes. Constructing 3D scene models from video for this process is difficult, due to the complex traffic conditions such as moving background and changing foreground objects. We develop the Tour Into the Traffic Video scheme (TITV) including mainly 4 stages: foreground traffic elements segmentation, background scene inpainting, scene models construction and walkthrough images rendering. For the first stage, we improved the Fast Two Cycle (FTC) level set method based on narrow band superpixels to automatically extract foreground traffic elements in the video sequence. A competition term based on maximum a posteriori is also integrated into the FTC level set framework. After the segmentation stage, the optical flows in occlusion foreground regions are inpainted by BP neutral networks and Gaussian regression process to keep their continuity. Gaussian Mixture Model is applied to inpaint the background since the pixels are corresponded by inpainted optical flows. Control points on road edges are specified to construct our TITV scene models. The 3D scene structure is constructed for each frame based on the control points. Foreground models such as vehicles and traffic signs are projected onto the background scene models. Users can change their viewpoints according to their own interpretations. Panoramic mode and non-photorealistic mode can also be switched. A database is constructed to represent the connection relationships of traffic elements for each frame. The touring trajectory is shown on the maps since we have GPS coordinates. Comprehensive user studies and comparisons well demonstrate the effectiveness of the proposed framework for giving user touring experience.

Index Terms- level set segmentation, maximum a posteriori, superpixel, optical flow, Gaussian regression, Gaussian mixture model, scene model, new view point.

1 Introduction

Maps with street photos have facilitated everyone's life. Google street view [1] [2] was embedded in google maps since 2006. With Google street view, users can tour into the spherical panoramic scenes by clicking on the. Microsoft company developed street slide [3] in 2010, which can let users switch between immersive panoramas and multi-perspective "strip" panoramas. Similar applications include Microsoft Phtosynth[4], QuickTime VR[5] of Apple Company. These applications are based on discrete photos captured from fixed view points of streets.

Autonomous driving systems will play an important role in the future traffic. When developing unmanned cars, it is expensive and time consuming to test the algorithms on real traffic road. Street photos based on the maps can be used for indoor test. In order for the unmanned cars to move freely, new viewpoint images generation becomes an important question.

New view point image generation are classified into 3 groups: 1) Depth-map based. In [6], Chen el. al. corresponded original images by depth maps to synthesize new viewpoint images. Their method is accelerated by a quadtree decomposition and a view-independent visible priority. Zitnick et. al. [7] used a color segmentation-based stereo algorithm to generate high-quality photoconsistent correspondences across all camera views. Mattes for areas near depth discontinuities are then extracted to reduce artifacts during view synthesis. 2)Single image based. Saxena et al. [8] generate 3D scene structure based on single still image and generated new view point images according to different projections angles based on texture mapping. Tour Into the Picture (TIP) [9] proposed by Horry et al. is an image-based method for generating a sequence of walk-through images from a single reference image. Kang et. al. [10] used vanishing line instead of vanishing point, and extend the concept of TIP to image sequence. 3) Multi-images based. Tanimota el al. [11]constructed the scheme of free-viewpoint television (FTV) based on the concept of ray space which captured by camera arrays.

Traffic video sequence captured from cameras mountained on moving cars contains much useful information of the world. The background of the video changes from frame to frame. The foreground objects, e.g. vehicles change their locations and shapes as well. We propose our Tour Into the Traffic Video (TITV) scheme to the traffic video sequence,

which mainly include 4 successive stages: foreground traffic elements segmentation, background scene inpainting, scene models construction and walkthrough images generation.

In the first stage, we use powerful level set method [12]-[14] to locate the contours of foreground objects for its many advantages, such as the automatic handling of topological changes. In the background inpainting stage, optical flows in occluded foreground regions are inpainted by background regions of adjacent frames. Back-propagation (BP) neural networks[33] and Gaussian regression[32][22] are used to inpaint the optical flow. Gaussian mixture model is then applied to inpaint the occluded pixels in the moving foreground regions by the predicted optical flow. In the scene models construction stage, the inpainted background images are projected to the background models according to the control points. The scene model is mainly composed of road plane, left wall plane and right wall plane. We adopt *RGBA* data structure for the extracted foreground images. The constructed foreground models are assumed to stand vertically in the scene model, e.g. vehicles on the road plane and traffic signs on the sky. Scene models for each frame are corresponded by the specified control points. After the scene model construction stage, we can render the walkthrough images. Special effects can be generated by non-photorealistic rending[35][36]. We have two scene model modes: traditional model and panoramic model. Users can switch between these two modes according to their own need.

The main contributions of this work are summarized as follows:

1) A superpixel based switching method and MAP approach is seamlessly integrated into the fast two-cycle (FTC) level set segmentation method.
2) An optical flow inpainting algorithm is proposed using BP neutral networks and Gaussian process regression. Gaussian Mixture models are used for the further image inpainting.
3) A novel TITV system is developed to give users touring experience of traffic scenes. Our scene model is also fit for panoramic traffic images. Special effects can be generated by non-photorealistic rendering. A database to represent the attachment relationship of traffic elements is also developed.

Fig. (1) shows our TITV flow diagram. The input data to our system are traffic image sequence (including panoramic images) with GPS coordinates. After foreground objects segmentation and background image inpainting stages, scene models can be constructed. Users can select their viewpoint and touring speed by steering wheel or joystick. The touring trajectory is then displayed on the map.

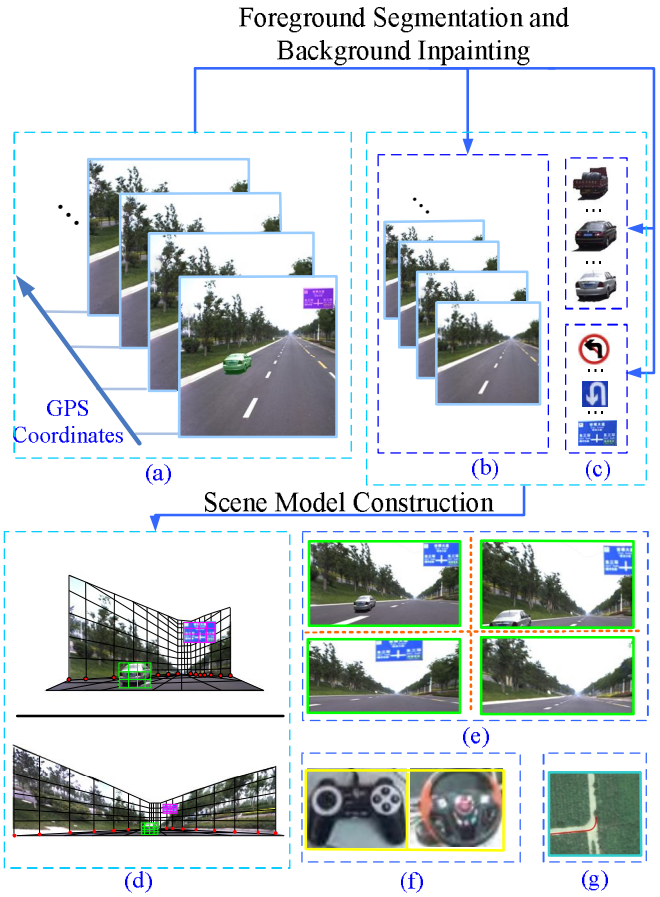


Figure 1: Flow diagram of TITV method. (a) input traffic video with GPS coordinates. (b) background image sequence. (c) foreground objects. (d) constructed scene models of current frame, upper: normal model; lower: panoramic model. (e) new view point images at time t. (f) user interaction input. (g) trajectory on the map.

2 Related Works

2.1 level-set based image segmentation

One related research direction to our work is level-set based foreground traffic elements segmentation. The goal of foreground segmentation is to extract the foreground vehicles and traffic signs in the traffic video sequence using spatial-temporal information. These foreground objects can further generate the foreground models in our TITV system.

Many approaches update the level-set function globally over the entire grid such as the Chan-Vese (CV) image segmentation model [15]. In order to reduce the computational cost for level-set methods, the level-set function is reinitialized by solving a Hamilton-Jacobi PDE for a fixed number of steps at every iteration of the evolving level-set function in [16]. In [17], the evolution of CV model is separated into two different cycles. Gi bou et al. draw a connection between level set algorithm and k-Means plus nonlinear diffusion preprocessing. Their method retains spatial coherence on initial data characteristic of curve evolution techniques. Shi et al. [18] proposed the fast two-cycle (FTC) level set method without solving PDEs. With that method, the evolution of level set curve is realized by switching mechanism between two linked lists of pixels. The curve evolution process is composed of two different cycles: one cycle for the data dependent term and a second cycle for the smoothness regularization.

The data structure of the FTC level set method is quite simple and consists of:

(a) an integer array $\mathbf{\Phi}(\mathbf{x})$ for the level-set function;

(b) an integer array $\mathbf{F}$ for the speed function;

(c) two lists of grid points adjacent to the evolving curve C: $\mathbf{L}_{in}$ and $\mathbf{L}_{out}$.

$\mathbf{L}_{in}$ and $\mathbf{L}_{out}$ for the object region $\mathbf{\Omega}$ are defined as follows:

$$\begin{aligned} \mathbf{L}_{in} &= \{\mathbf{x} \mid \mathbf{\Phi}(\mathbf{x}) < 0 \; and \; \exists \mathbf{y} \in N_4(\mathbf{x}) \; such \; that \; \mathbf{\Phi}(\mathbf{y}) > 0\}, \\ \mathbf{L}_{out} &= \{\mathbf{x} \mid \mathbf{\Phi}(\mathbf{x}) > 0 \; and \; \exists \mathbf{y} \in N_4(\mathbf{x}) \; such \; that \; \mathbf{\Phi}(\mathbf{y}) < 0\} \end{aligned} \tag{1}$$

switchIn() and *switchOut*() functions are defined for curve evolution. Both the functions are determined by the speed function $\mathbf{F}$. *switchIn*() function is applied if the pixels in $\mathbf{L}_{out}$ have larger probability belong to foreground than background, while *switchOut*() function is applied if the pixels in $\mathbf{L}_{in}$ have larger probability belong to background than foreground. These two functions are implemented continuously until the stopping condition is satisfied.

2.2 Background Scene Inpainting based on Optical Flow

Background images without moving foreground objects are the precondition for background model construction. Many research works are based on static background inpainting. Hsu et. al. [19] applied a hierarchical block-based technique to construct

mosaics from video sequences with moving objects. They estimated the global motion and exclude the moving objects from the video mosaic. Kang et. al. [10] used camera calibration to correspond pixels from successive image sequences. The corresponded pixels can be considered as a stochastic process, thus Gaussian Mixture Models can be applied to predict background pixels. However, this method fits for the condition where a camera rotating in a fixed scene periodically and precise camera calibration is demanded. Wexler et. al. [20] proposed methods to fill in the missing portions in video by sampling spatio-temporal patches from available parts of the video, while enforcing global spatio-temporal consistency between all patches in and around the hole. Their method is also designed for static background videos.

Optical flow is useful information to correspond pixels of adjacent video frames. Since the optical fields of moving foregrounds differ from backgrounds, we apply optical flow inpainting method for the occluded foregrounds. Acker et. al. [21] proposed method to reconstruct optical flow at positions indicated by confidence measures using inpainting. Kim et. al[22] use Gaussian process regression to represent the motion tendency as a stochastic vector field. In our experiment, we combined BP neutral networks and Gaussian Process Regression to inpaint optical flows, and then use Gaussian Mixture Models to complete the occluded foreground regions since the pixels are corresponded by inpainted optical flows.

2.3 Scene Model Construction and Rendering

Saxena et al. [8] proposed methods for each super pixel in the image, markov random field is applied to judge the plane parameters to construct 3D scene model. Hoiem et.al. [23] use support vector machine (SVM) to cluster superpixels into constellations. Based on superpixel constellations, they further infer ground and sky regions to recover precise geometry.

Horry et. al. [9] proposed the concept of Tour Into the Picture (TIP) scheme according to the vanishing point of an image, which is to construct a simplified 3D scene model composing of left wall, right wall, back wall, sky and plane. By navigating a 3D scene model constructed from the image, TIP provides convincing 3D effects. Similar methods were described in [39][41]. Kang et al. modified the TIP scheme based on vanishing line [24], and extend it to video sequence[10]. However, this method fits for 2 limited

conditions: 1) a static camera 2) a camera rotates periodically. Faced with these limitations, we proposed our TITV model which deals with successive image sequences captured by a camera arranged on a moving vehicle.

Adelson et. al. [25] proposed the plenoptic function to model the world from a certain view point. That is, a 7-dimensional function composed of observing angle, view point, time and the wavelength of light. The plenoptic function can be used to project 3D scene model to new viewpoint image plane.

3 Foreground Segmentation based on Level-set Method

Fig. 2. shows the flow diagram of our level set based foreground detection process. The detection process mainly includes 2 stages. The first stage is the narrow band two-cycle level set segmentation based on superpixel feature pools. The second stage is a refining process using maximum a posteriori (MAP) estimation. The segmentation result of each frame is used to update feature pools for the next frame. This method can be used to extract foreground traffic elements including moving vehicles, traffic signs and traffic lights.

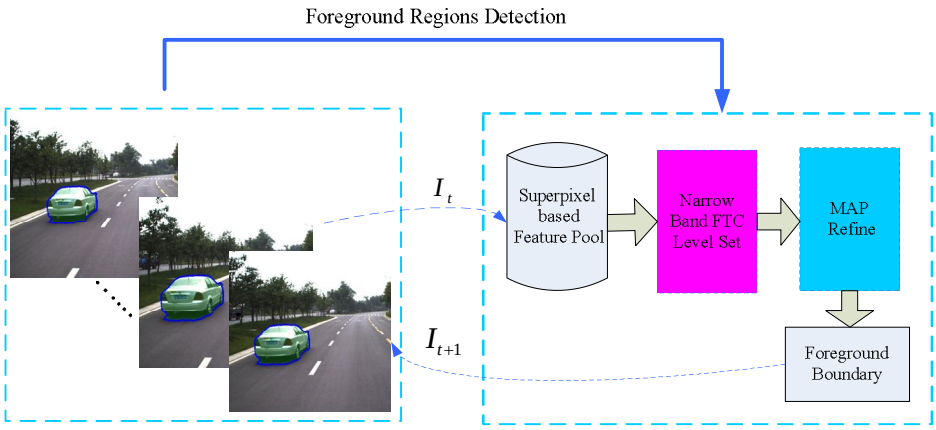


Figure 2: Flow diagram of our level set based foreground segmentation.

3.1 FTC level set based on narrow band perception of background

Motivated by Mumford-Shah functional and the level set function based on it[17][26], we define an energy functional as follows:

$$E_1 = \underbrace{\mu\int_\Omega |\nabla H(\mathbf{\Phi})| d_\Omega}_{E_s} \underbrace{-\lambda_1\sum_{m=1}^{M}\int_\Omega p(\mathbf{f}(\mathbf{x})|\Omega_m)H(\mathbf{\Phi})d_\Omega - \lambda_2\sum_{b=1}^{B}\int_\Omega p(\mathbf{f}(\mathbf{x})|\Omega_b)(1-H(\mathbf{\Phi}))d_\Omega}_{E_d} \quad (2)$$

where E_d is the data fidelity term that represents the likelihood of the current scene, E_s is for smoothness regularization and is proportional to the length of all curves. $\mathbf{f}(\mathbf{x})$ is the feature vector defined at pixel location $\mathbf{x}$. H is the Heaviside distribution. Assuming there are M object regions. $\mathbf{\Omega}_m$ is the $m-th$ foreground region, $\mathbf{\Omega}_b$ is the $m-th$ narrow band background region around $\mathbf{\Omega}_m$ with width of ω_{NB}. $\mathbf{\Omega}_b$ is defined by:

$$\mathbf{\Omega}_b = \{\mathbf{x} \in \overline{\Omega_m} \mid \min_{\mathbf{y} \in \Omega_m} \|\mathbf{x} - \mathbf{y}\| \le \omega_{NB}\}. \tag{3}$$

The region competition term [38] between foreground and background is defined by:

$$F_d = \log \frac{p(\mathbf{f}(\mathbf{x}) \mid \Omega_m)}{p(\mathbf{f}(\mathbf{x}) \mid \Omega_b)} \qquad \text{s.t.} \quad \mathbf{f}(\mathbf{x}) = \omega_1 \bullet \mathbf{f}_C(\mathbf{x}) + \omega_2 \bullet \mathbf{f}_T(\mathbf{x}) \tag{4}$$

$\mathbf{f}(\mathbf{x})$ is composed of 2 parts: color feature $\mathbf{f}_C(\mathbf{x})$ and texture feature $\mathbf{f}_T(\mathbf{x})$. ω_1 and ω_2 are the weight coefficients to obtain a balance between the two features.

The color feature distribution can be measured either based on superpixels or K-Means cluster in CIE-LAB color space, as shown in Fig. 3(c). We choose superpixel based color distribution. The superpixels are segmented using Mori's method [27]. Fig. 3(a) shows the image segmentation results into superpixels of 32, 64, 256 size pixels (approximately). We choose 64 pixel size to generate superpixel feature pools for foreground and background region:

$$\mathbf{Color}_{fg} = \{\mathbf{SP}_r \mid r = 1, ..., N_m\}, \mathbf{Color}_{bg} = \{\mathbf{SP}_r \mid r = 1, ..., N_b\} \tag{5}$$

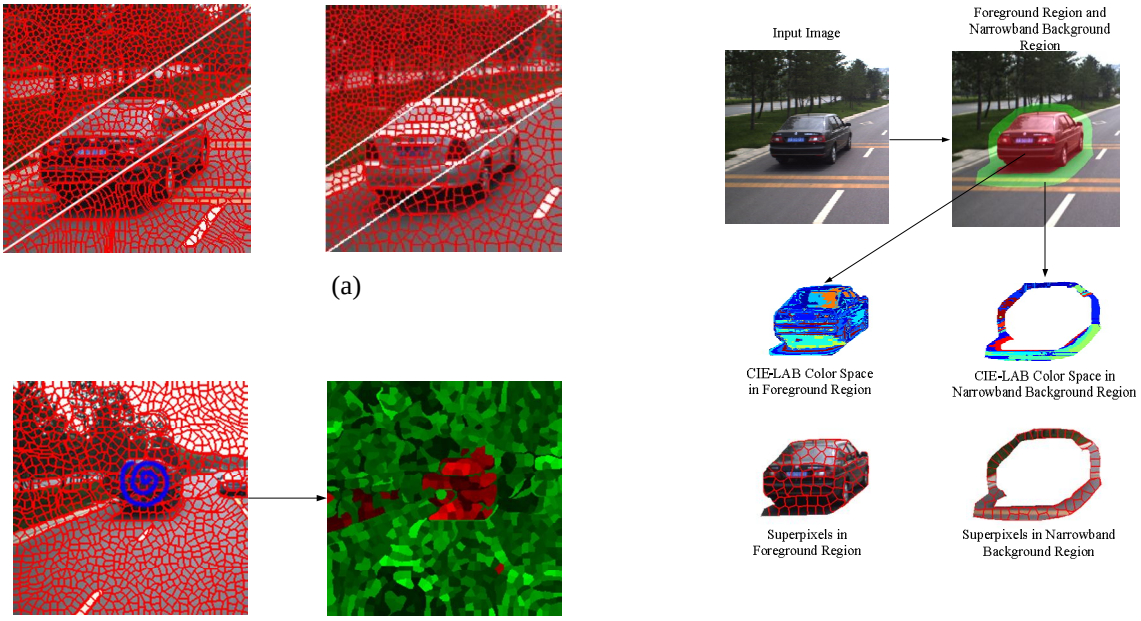
where N_m and N_b are the superpixel numbers in foreground and narrow band background regions respectively.

Kmeans clustering of superpixel histograms are applied to obtain the texture feature pools:

$$\mathbf{Texture}_{fg} = \{\mathbf{M}_{fg}\{i\} \mid i = 1, ..., K_m\}, \ \mathbf{Texture}_{bg} = \{\mathbf{M}_{bg}\{i\} \mid i = 1, ..., K_b\} \tag{6}$$

$\mathbf{M}_{fg}\{i\}$ is the $i-th$ mean center of foreground texture pool, $\mathbf{M}_{bg}\{i\}$ is the $i-th$ mean center of narrow band background texture pool. K_m and K_b are cluster numbers respectively.

The initial curve is extracted from the target-background saliency map according to user stroke (Fig. 3(b))[40]. The saliency map is measured by Bhattacharyya distance of histograms.



(a) (b) (c)

Figure 3: Superpixels and narrow band perception of background. (a) images segmented into superpixels of different sizes. (b) the computed saliency map of superpixels according to the stroke. (c) foreground and narrow band background based on CIE-LAB color space and superpixels.

For color feature part, we have definitions as:

$$p(\mathbf{f}_C(\mathbf{x})\,|\,\Omega_m) = \sup_{SP\{i\}\in\Omega_m}\{p(\mathbf{f}_C(\mathbf{x})\,|\,SP\{i\}\},\ p(\mathbf{f}_C(\mathbf{x})\,|\,\Omega_b) = \sup_{SP\{i\}\in\Omega_b}\{p(\mathbf{f}_C(\mathbf{x})\,|\,SP\{i\}\} \quad (7)$$

where $p(\mathbf{f}_C(\mathbf{x})\,|\,SP\{i\})$ denotes the color distribution based on the $i-th$ superpixel, according to its color mean and variance.

For texture feature part, we defined:

$$p(\mathbf{f}_T(\mathbf{x})\,|\,\Omega_m) = \sup_{\mathbf{M}_{fg}\{i\}\in\mathbf{Texture}_{fg}} \exp(\{r_{\mathbf{M}_{fg}\{i\}}\}),\quad p(\mathbf{f}_T(\mathbf{x})\,|\,\Omega_b) = \sup_{\mathbf{M}_{bg}\{i\}\in\mathbf{Texture}_{bg}} \exp(\{r_{\mathbf{M}_{bg}\{i\}}\}) \quad (8)$$

where $\mathbf{f}_T(\mathbf{x})$ is $N_T \times N_T$ block centered at $\mathbf{x}$. $r_{\mathbf{M}_{fg}\{i\}}$ is the average correlation coefficient between $\mathbf{f}_T(\mathbf{x})$ and the $i-th$ cluster center of $\mathbf{Texture}_{fg}$. $r_{\mathbf{M}_{bg}\{i\}}$ is that between $\mathbf{f}_T(\mathbf{x})$ and the $i-th$ cluster center of $\mathbf{Texture}_{bg}$.

The average correlation coefficient is computed by:

$$r_{\mathbf{M}\{i\}} = \frac{N\sum_{j=1}^{N}\mathbf{f}_T(\mathbf{x})[j]\bullet M\{i\}[j] - \sum_{j=1}^{N}\mathbf{f}_T(\mathbf{x})[j]\bullet\sum_{j=1}^{N}\mathbf{M}\{i\}[j]}{\sqrt{(N\sum_{j=1}^{N}\mathbf{f}_T{}^2(\mathbf{x})[j]) - (\sum_{j=1}^{N}\mathbf{f}_T(\mathbf{x})[j])^2)\bullet(N\sum_{j=1}^{N}\mathbf{M}^2\{i\}[j]) - (\sum_{j=1}^{N}\mathbf{M}\{i\}[j])^2)}} \tag{9}$$

where N is the dimension of histogram.

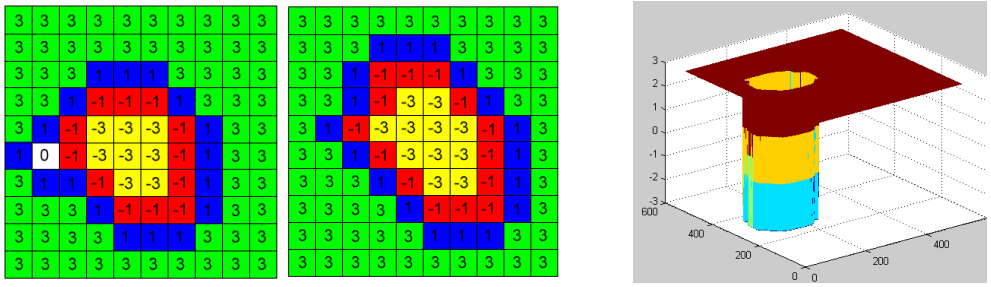
So (4) can be rewritten as:

$$F_d = \frac{\omega_1 \bullet \sup_{SP\{i\}\in\mathbf{Color}_{fg}}\{p(\mathbf{f}_C(\mathbf{x})\,|\,SP\{i\}\} + \omega_2 \bullet \sup_{\mathbf{M}_{fg}\{i\}\in\mathbf{Texture}_{fg}}\exp(\{r_{\mathbf{M}_{fg}\{i\}}\})}{\omega_1 \bullet \sup_{SP\{i\}\in\mathbf{Color}_{bg}}\{p(\mathbf{f}_C(\mathbf{x})\,|\,SP\{i\}\} + \omega_2 \bullet \sup_{\mathbf{M}_{bg}\{i\}\in\mathbf{Texture}_{bg}}\exp(\{r_{\mathbf{M}_{bg}\{i\}}\})} \tag{10}$$

We define two lists of neighboring pixels $\mathbf{L}_{in}$ and $\mathbf{L}_{out}$. Besides the inside and outside neighboring pixels contained in $\mathbf{L}_{in}$ and $\mathbf{L}_{out}$, we call those pixels inside $\mathbf{C}_1$ but not in $\mathbf{L}_{in}$ as interior pixels and those pixels outside $\mathbf{C}_1$ but not in $\mathbf{L}_{out}$ as exterior pixels.

$\mathbf{\Phi}$ matrix is defined as follows :

$$\mathbf{\Phi}(\mathbf{x}) = \begin{cases} 3 & if\ \mathbf{x}\ is\ an\ exterior\ pixel \\ 1 & if\ \mathbf{x}\in\mathbf{L}_{out} \\ 0 & if\ \mathbf{x}\ is\ an\ ambiguous\ pixel \\ -1 & if\ \mathbf{x}\in\mathbf{L}_{in} \\ -3 & if\ \mathbf{x}\ is\ an\ interior\ pixel \end{cases} \tag{11}$$



(a) (b)

Figure 4: Indication of the Level-set $\mathbf{\Phi}$ matrix. (a) the evolution of the boundary pixels from t to $t+1$. 3 and -3 denote the exterior pixels and interior pixels respectively. 1 and -1 denote the pixels in $\mathbf{L}_{out}$ and $\mathbf{L}_{in}$. 0 is the ambiguous pixel. (b) 3D visualization of $\mathbf{\Phi}$ at time t.

We also allow 0 to exist between $\mathbf{L}_{in}$ and $\mathbf{L}_{out}$, in case the pixels in $\mathbf{L}_{in}$ and $\mathbf{L}_{out}$ are close in color and texture. We call these pixels ambiguous pixels. The ambiguous pixels

can be assigned weights according to fuzzy math [28]. The *switchIn*() and *switchOut*() are modified in that the ambiguous pixels are used as a buffer if a pixel switch from $\mathbf{L}_{out}$ to $\mathbf{L}_{in}$ (or from $\mathbf{L}_{in}$ to $\mathbf{L}_{out}$). Fig. 4. visualizes $\mathbf{\Phi}$ matrix in 2D and 3D.

The switching mechanism is determined by the sign of F_d rather than solving PDE functions. Table I shows the two cycle switching process. The stopping condition is either the following:

(a) $F_d(\mathbf{x}) \le 0 \ \forall \mathbf{x} \in \mathbf{L}_{out}$ and $F_d(\mathbf{x}) \ge 0 \ \forall \mathbf{x} \in \mathbf{L}_{in}$.
(b) A pre-specified maximum iteration number reached.

Table I

Two Cycle Switching Process

- **Step 1:**
- Compute the speed F_d for each pixel in $\mathbf{L}_{out}$ and $\mathbf{L}_{in}$.
- For each pixel $\mathbf{x} \in \mathbf{L}_{out}$ (or $\mathbf{x} \in \mathbf{L}_{in}$), *switchIn*($\mathbf{x}$) (or *switchOut*($\mathbf{x}$)) according to the sign of $F_d^{'}(\mathbf{x})$. $\mathbf{\Phi}(\mathbf{x})$ is also updated. $\forall$ pixel $\mathbf{x} \in \mathbf{L}_{in}$ (or $\mathbf{x} \in \mathbf{L}_{out}$) satisfy all its neighboring pixels in $\mathbf{\Phi}$ are negative (or positive), delete it from $\mathbf{L}_{in}$ (or $\mathbf{L}_{out}$).
- Check the stopping condition. If it is satisfied, go to step 2; otherwise continue this step.
- **Step 2:**
- For each pixel $\mathbf{x} \in \mathbf{L}_{out}$ and $\mathbf{L}_{in}$, Gaussian filter of size $N_G \times N_G$ as $\mathbf{G}$ is applied to incorporate boundary smoothness regularization.
- Check the stopping condition. If it is not satisfied, continue this step.

Table II shows our narrow band superpixel based FTC level set method. Ambiguous pixels were used as buffers for *switchIn*() and *switchOut*() processes.

Table II

FTC Level Set based on Narrow Band Superpixels

- **Stage 1: Initialization Stage**
  - Segment each frame into superpixels.
  - Compute the target-background confidence map for the first frame. Initialize $\mathbf{L}_{in}, \mathbf{L}_{out}$, $\mathbf{\Phi}$, in the first frame according to the confidence map.
  - Obtain the superpixel based feature pool for the first frame, which are composed of color and texture features:

    $\mathbf{Color}_{fg} = \{\mathbf{SP}_r^t \mid t = 1; r = 1, ..., N_m\}, \mathbf{Color}_{bg} = \{\mathbf{SP}_r^t \mid t = 1; r = 1, ..., N_b\}$

    $\mathbf{Texture}_{fg} = \{\mathbf{M}_{fg}^t\{i\} \mid t = 1; i = 1, ..., K_m\}$, $\mathbf{Texture}_{bg} = \{\mathbf{M}_{bg}^t\{i\} \mid t = 1; i = 1, ..., K_b\}$
  - F_d is defined according to (10).
- **Stage 2: Tracking Stage**

  for $t = 2$ to the end of the sequence
  - Two cycle switching process(Table I).
  - Save the two linked lists $\mathbf{L}_{out}$, $\mathbf{L}_{in}$, $\mathbf{\Phi}$ for further refine.
  - Update the color feature pool and texture feature pool at time t.

  end

3.2 Maximum A Posteriori Optimization

Due to the complexity of traffic environment, the segmentation results of section 3.1need to be refined. We further apply the Maximum A Posteriori (MAP) framework introduced by Mansouri[29] to optimize our curves. The MAP method uses Baye's theorem and assumes conditional independence between image pixels:

$$\begin{aligned} \mathbf{\Omega}_m^{t\,*} &= \arg\max_{\mathbf{\Omega}_m^t} p(\mathbf{\Omega}_m^t \mid \mathbf{I}_{t-1}, \mathbf{I}_t, \mathbf{\Omega}_m^{t-1}) \\ &= \arg\max_{\mathbf{\Omega}_m^t} \prod_{\mathbf{x}\in\mathbf{D}} p(\mathbf{I}_t(\mathbf{x}) \mid \mathbf{I}_{t-1}, \mathbf{\Omega}_m^{t-1}, \mathbf{\Omega}_m^t)\, p(\mathbf{\Omega}_m^t \mid \mathbf{I}_{t-1}, \mathbf{\Omega}_m^{t-1}) \end{aligned} \tag{12}$$

where $\mathbf{\Omega}_m^t$ and $\mathbf{\Omega}_m^{t-1}$ are $m-th$ foreground regions of current frame and previous frame. $\mathbf{I}_t$ and $\mathbf{I}_{t-1}$ denote current frame and previous frame. $\mathbf{D}$ is the image domain.

Probability $p(\mathbf{I}_t(\mathbf{x}) \mid \mathbf{I}_{t-1}, \mathbf{\Omega}_m^t, \mathbf{\Omega}_m^{t-1})$ is the likelihood of observing a particular color at space-time location $(\mathbf{x}, t)$ given current image and both current and next object regions. Prior probability $p(\mathbf{\Omega}_m^t \mid \mathbf{I}_{t-1}, \mathbf{\Omega}_m^{t-1})$ models available prior knowledge about object shape and/or motion.

Our second energy functional is thus the following:

$$\begin{aligned} E_2 = & \underbrace{-\int_{\Omega_m^t} \log p_t^{in}(\mathbf{I}_t(\mathbf{x}) \mid \mathbf{I}_{t-1}, \mathbf{\Omega}_m^{t-1}, \mathbf{\Omega}_m^t) d_{\mathbf{x}}}_{posterior\ in\ foreground} \underbrace{-\int_{\Omega_b^t} \log p_t^{NB}(\mathbf{I}_t(\mathbf{x}) \mid \mathbf{I}_{t-1}, \mathbf{\Omega}_m^{t-1}, \mathbf{\Omega}_m^t) d_{\mathbf{x}}}_{posterior\ in\ narrow\ band\ background} \\ & \underbrace{-\int_{\Omega_{fb}^t} \log p_t^{FB}(\mathbf{I}_t(\mathbf{x}) \mid \mathbf{I}_{t-1}, \mathbf{\Omega}_m^{t-1}, \mathbf{\Omega}_m^t) d_{\mathbf{x}}}_{posterior\ in\ far\ away\ background} \underbrace{-\log p(\mathbf{\Omega}_m^t \mid \mathbf{I}_{t-1}, \mathbf{\Omega}_m^{t-1})}_{prior\ probability} \end{aligned} \tag{13}$$

Since we have got a coarse curve in section 3.1, we only use the competition between the first two terms :

$$F_d^{'} = \log \frac{p^{in}(\mathbf{I}_t(\mathbf{x}) \mid \mathbf{I}_{t-1}, \mathbf{\Omega}_{t-1}^m, \mathbf{\Omega}_t^m)}{p^{NB}(\mathbf{I}_t(\mathbf{x}) \mid \mathbf{I}_{t-1}, \mathbf{\Omega}_{t-1}^m, \mathbf{\Omega}_t^m)} \tag{14}$$

where

$$\begin{aligned} p^{in}(\mathbf{I}_t(\mathbf{x}) \mid \mathbf{I}_{t-1}, \mathbf{\Omega}_m^{t-1}, \mathbf{\Omega}_m^t) \approx & \sup_{\mathbf{y} \in \Omega_m^{t-1}} \{\exp(-\left\|\mathbf{I}_t(\mathbf{x}) - \mathbf{I}_{t-1}(\mathbf{y})\right\|^2 (2\sigma_b^2)^{-1}) \\ & \bullet \exp(-\left\|(\mathbf{x}-\mathbf{y}) - \mathbf{v}(\mathbf{y})\right\|^2 (2\sigma_v^2)^{-1})\} \\ p^L(\mathbf{I}_t(\mathbf{x}) \mid \mathbf{I}_{t-1}, \mathbf{\Omega}_m^{t-1}, \mathbf{\Omega}_m^t) \approx & \sup_{\mathbf{y} \in \Omega_b^{t-1}} \{\exp(-\left\|\mathbf{I}_t(\mathbf{x}) - \mathbf{I}_{t-1}(\mathbf{y})\right\|^2 (2\sigma_b^2)^{-1}) \\ & \bullet \exp(-\left\|(\mathbf{x}-\mathbf{y}) - \mathbf{v}(\mathbf{y})\right\|^2 (2\sigma_v^2)^{-1})\} \end{aligned} \tag{15}$$

Assuming the non-rigid transformation from $\mathbf{I}_t$ to $\mathbf{I}_{t+1}$ can be expressed with motion field $\mathbf{v}$ and an additive white Gaussian noise $\mathbf{b}$ [31]:

$$I_t(\mathbf{x}+\mathbf{v}(\mathbf{x})) = I_{t-1}(\mathbf{x}) + \mathbf{b}(\mathbf{x}) \qquad \text{s.t.}\, \mathbf{b} \sim N(0:\sigma_b^2),\ \ \mathbf{v} \sim N(\mathbf{v}_O(x); \sigma_v^2) \tag{16}$$

where $\mathbf{v}_O(\mathbf{x})$ is the optical flow between $\mathbf{I}_{t-1}$ and $\mathbf{I}_t$ at position $\mathbf{x}$ according to Bruhn's method [30].

Table III

Boundary Refine Stage

- **Stage 3: Refine Stage**

for $t = 2$ to the end of the sequence

- Load $\mathbf{L}_{out}$, $\mathbf{L}_{in}$, $\mathbf{\Phi}$ computed from Table II.
- Compute the corresponding optical flow map from $\mathbf{I}_t$ to $\mathbf{I}_{t-1}$. Obtain $\mathbf{v}_{OP}(\mathbf{x})$ at each pixel location $\mathbf{x}$.
- F_d is defined according to (14).
- Two cycle switching process(Table I).

end

If we do not rely on the FTC framework, we can use basic greedy algorithm to minimize E_2. In this condition, we can define:

$$p(\mathbf{R}_{t+1} \mid \mathbf{I}_t, \mathbf{I}_{t+1}, \mathbf{R}_t) = \exp\{-d(f(\mathbf{s}_{n,RC}), f(\mathbf{s}_{n+1,RC}))\} \qquad \text{s.t. } f(\mathbf{s}_{n,RC}) = \sum_{j=1}^{N_{RC}} K\left(\frac{\mathbf{s}_{n,RC} - \mathbf{s}_{n,j}}{h}\right) \tag{17}$$

where $\mathbf{s}_{n,RC}$ is the feature vector of the region centroid $\mathbf{x}_{n,RC}$, $\mathbf{s}_{n,j}$ is the feature vector of the pixels located within the foreground region, and N_{RC} is number of pixels of this region. K is the Epanechnikov kernel, ρ is the Bhattacharyya coefficient.

3.3 Segmentation Results Evaluation

Our FTC level set algorithms were implemented on a HP Personal Computer with 2.0 GHZ AMD process and 2.0G RAM using MATLAB R14. Four typical image sequences “Red car” (512x512x100), “Black car” (512x512x100), “White car” (512x512x50) and “Traffic sign” (256x256x60) were used in our experiments.

Fig. 5 shows some tracking results based on narrow band superpixel FTC described in §3.1. The blue line and red line denote the two linked lists of contours($\mathbf{L}_{out}$ and $\mathbf{L}_{in}$). Fig. 6 compares the tracking results of FTC level set with MAP refine(§3.2), FTC level set with narrow band superpixels, and the orginal FTC level set. The original FTC level set was implemented in gray scale images. We can see more accurate curves were obtained through refined FTC with MAP.

Table IV quantitatively compare our algorithms against Shi's (orginal FTC level set) and Mansouri's level set method. The evaluation is based on the three quantitatively performance measures: Precision, Recall and F-score compared with the ground truth boundary curves. We can see from the table that our propose methods can improve the tracking performance significantly.

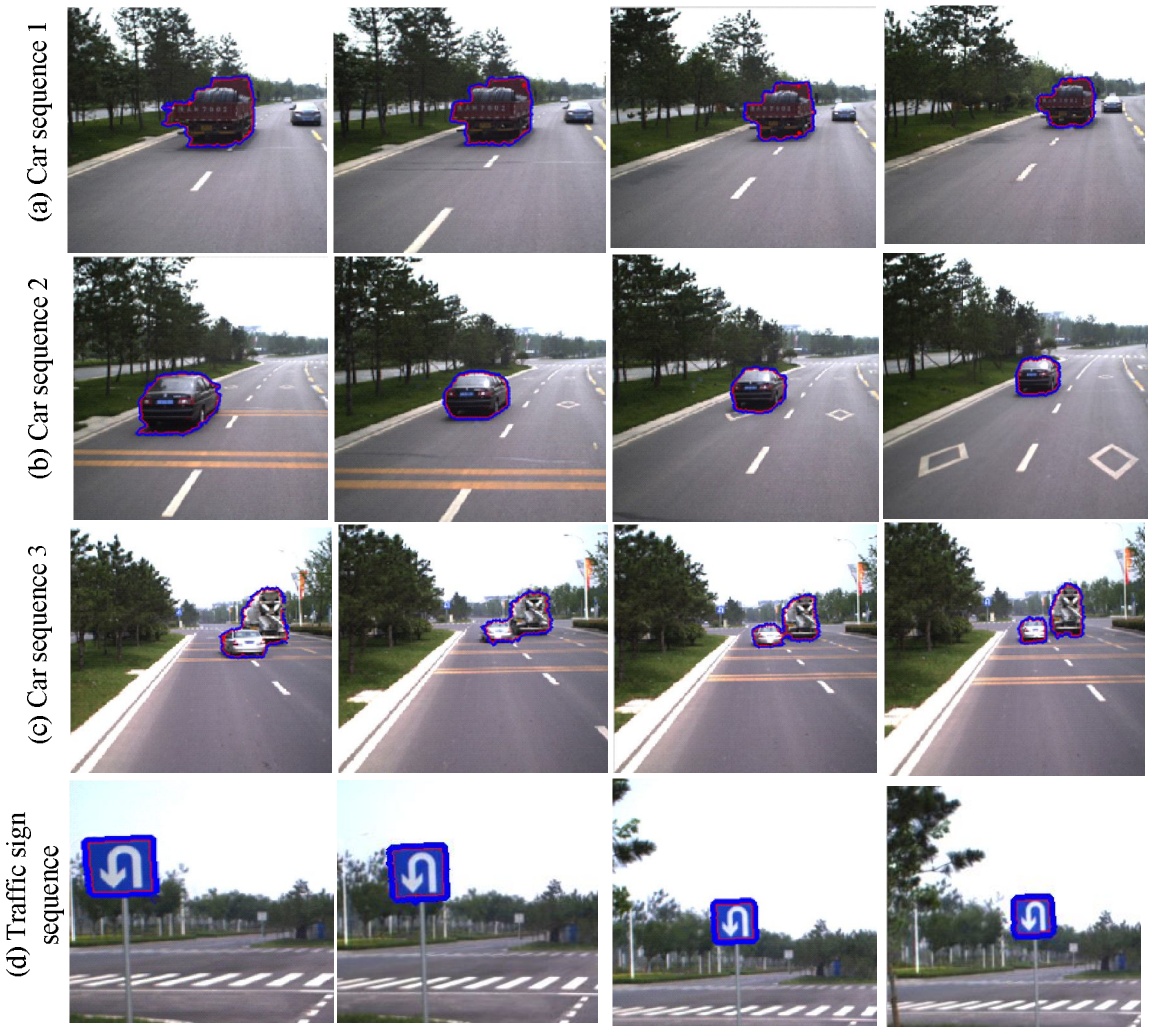
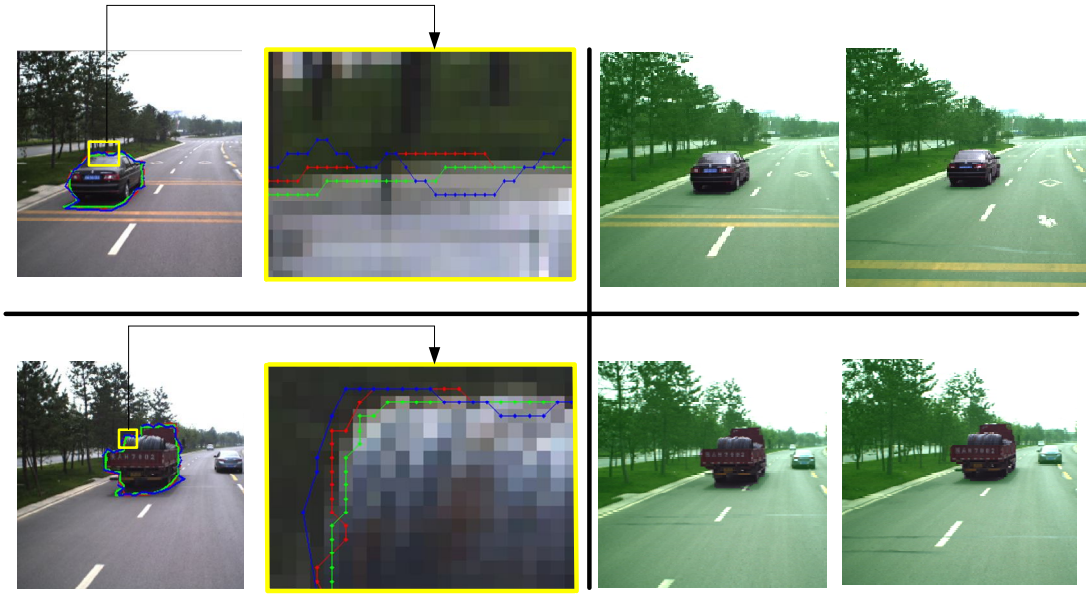


Figure 5: Segmentation results of TCL based on narrow band superpixels.



(a) (b)

Figure 6: Refined foreground region. (a) Comparison of the tracking results of our narrow band superpixel based FTC (red line) and MAP refined result (green line), and the orginal FTC based on gray scale image(blue line). We can see more accurate boundaries can be obtained through refined MAP method. (b) Some refined segmentation results based on MAP refined FTC.

TABLE IV

RESTULS OF QUANTITATIVE EVALUATION

	Methods	**Precision**	**Recall**	**F-score**
Red car	Shi's	0.8746	0.7955	0.8332
	Mansouri	0.9029	0.9093	0.9061
	Our FTC	0.9253	0.9300	0.9276
	Our FTC+MAP	0.9424	0.9411	**0.9417**
Black car	Shi's	0.8521	0.7849	0.8161
	Mansouri	0.9104	0.9243	0.9173
	Our FTC	0.9082	0.9200	0.9141
	Our FTC+MAP	0.9217	0.9349	**0.9280**

4 Background Scene Inpainting by Optical Flow

After the foreground regions of current frame is specified by the methods described in section 3, the pixels in these occlusion regions can be further inpainted by spatial-temporal information from adjacent frames. Our background scene inpainting process is composed of 2 successive parts: (1) optical flow inpainting and (2) background inpainting based on optical flow.

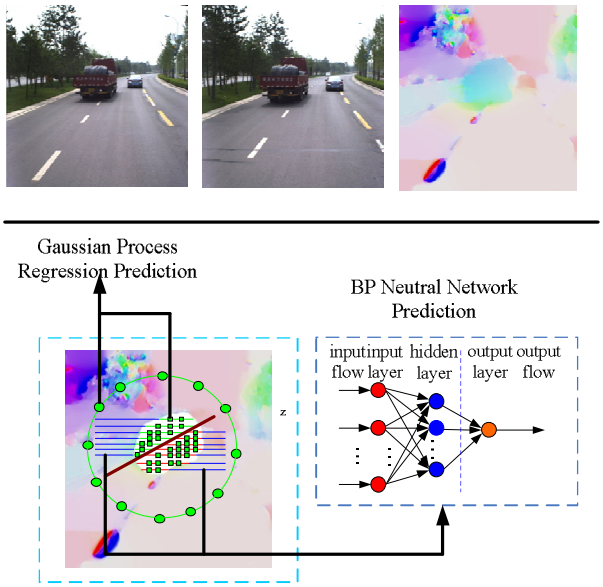


Figure 7: Our optical flow inpainting process.

4.1 optical flow inpainting

We computed optical flow from current frame to adjacent frames according to [30]. Fig. 7 (upper part) shows an example of optical flow map between adjacent frames. We can see optical flow discontinuities in magnitude and direction between foreground and background regions. Gaussian process regression and BP neutral networks are applied to inpaint the optical flows in occluded foreground regions. The inpainted flows can be used to correspond the occluded pixels to background regions in adjacent frames.

We assume the optical flows maintain continuity in each separate background region of the traffic scene photo, e.g. sky region and road region. The BP neutral network is used to predict occluded flows row by row, as shown in Fig. 7(lower part). Local extreme flows may exist if we only use the background information for prediction. So we propose using the Gaussian process regression model [22] [32] as an initialization step to predict anchor point (green squares inside the occlusion region) .

(1) Gaussian Process Regression as Initialization

We assume each velocity component at the location $\mathbf{x}$ follows the regression model $\hat{\mathbf{y}} = f(\mathbf{x}) + \varepsilon$, where $\varepsilon \sim N(0, \sigma_v^2)$, i.e., Normal distribution. Each location $\mathbf{x}$ has a set of noisy observed velocity vector components: $\mathbf{y}_u$ (the velocity component in the u-axis), $\mathbf{y}_v$ (the velocity component in the v-axis).

We sample training optical flows from background region of radius R (green circle). The training data set is defined by $\mathbf{T} = \begin{bmatrix} \mathbf{x}_1, & ..., & \mathbf{x}_N \\ y_1, & ..., & y_N \end{bmatrix}$. The $N \times N$ covariance matrix $\mathbf{K}$ is defined as $[\mathbf{K}]_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$, where

$$K(\mathbf{x}, \mathbf{x}^{"}) = E[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}^{"}) - m(\mathbf{x}^{"}))] = E[f(\mathbf{x}) f(\mathbf{x}^{"})]. \quad (18)$$

We then define the observation vector $\mathbf{y} = [y_1, ..., y_N]^T$; $\mathbf{y}$ can be shown as a zero mean multivariate Gaussian process with a covariance matrix $\mathbf{K}^* = \mathbf{K} + \sigma^2 \mathbf{I}$. The posterior density for a test point $\mathbf{x}^*$, $p(y^* \mid \mathbf{x}^*, \mathbf{T})$ is a univariate normal distribution with the mean $\bar{y}^*$ and the variance $\text{var}(\bar{y}^*)$

$$\bar{y}^* = \mathbf{k}(\mathbf{x}^*)^T (\mathbf{K}^*)^{-1} \mathbf{y}, \text{var}(\bar{y}^*) = K(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}(\mathbf{x}^*)^T (\mathbf{K}^*)^{-1} \mathbf{k}(\mathbf{x}^*) \quad (19)$$

s.t. $\mathbf{k}(\mathbf{x}^*) = [K(\mathbf{x}^*, \mathbf{x}_1), ..., K(\mathbf{x}^*, \mathbf{x}_n)]^T$

We use (19) to predict the optical flows for anchor points in each occlusion row, as shown in Fig. 7(green rectangles inside the occlusion region).

(2) BP Neutral Network Prediction

We apply 3-layerd BP neutral networks [33] to predict occluded optical flows $\mathbf{y}_u$ and $\mathbf{y}_v$ respectively for each row, as shown in the right part of Fig. (8). The training dataset is composed of 2 parts: (a) the anchor points computed by Gaussian regression model, and (b) the optical flows in the same row outside the occlusion region.

4.2 Background image inpainting based on optical flow

We constructed a Gaussian mixture model to the pixels corresponded by optical flow. For each pixel $\mathbf{x}$ on the background image $\mathbf{I}_B$, we keep track of its history:

$$\{\mathbf{s}_i : \mathbf{s}_i = \mathbf{I}_B(\mathbf{x}, i), 1 \le i \le k\} \quad (20)$$

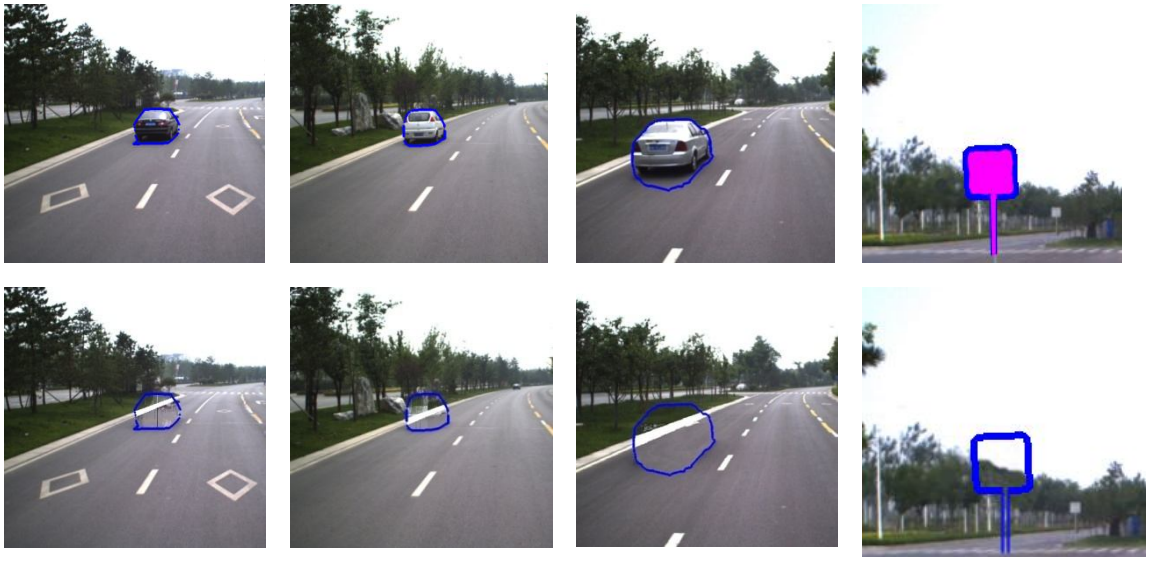
where $\mathbf{I}_B(\mathbf{x}, i)$ is the color value at $\mathbf{x}$ at the i_{th} adjacent frame in the sequence. The recent history of each pixel is modeled by a mixture of N Gaussian distributions and the probability of observing the current pixel value is

$$p(\mathbf{s}_i) = \sum_{j=1}^{N} \omega_{j,i} \eta(\mathbf{s}_i, \boldsymbol{\mu}_{j,i}, \sigma_{j,i}^2) \tag{21}$$

where N is the number of distributions, $\omega_{j,i}$ is an estimate of the weight of the j_{th} Gaussian in the mixture at time i, and where η_i is the corresponding Gaussian probability density function with a mean value $\boldsymbol{\mu}_{j,i}$ and a variance $\sigma_{j,i}^2$.

The Gaussian mixture model is updated each time a new adjacent frame is added. If none of the existing distributions match current value, a new distribution is constructed. After the k reference frames are compared, the distribution with the highest probability will be chosen as the color of the pixel location $\mathbf{x}$.

The road boundaries are specified in advance, so we apply this inpainting method to each separated regions. Fig. 8 shows some inpainting results:



(a) (b) (c) (d)

Figure 8: Our background inpainting results. (a) ,(b) and (c) used the method described in 4.1 and 4.2 for moving foregrounds. (d) is based on the method of 4.3, which deals with static foregrounds, e.g. traffic signs. Foreground mask is used here.

4.3 Image inpainting for static foregrounds

The methods described in section 4.1 and 4.2 can be used to inpaint the moving foregrounds such as vehicles. For the static foregrounds like traffic signs and traffic lights, we can use [42] to implement the inpainting based on foreground masks. Fig. 8 (d) shows the inpainting result.

5 Scene Structure and the Touring Experience

5.1 Scene model construction

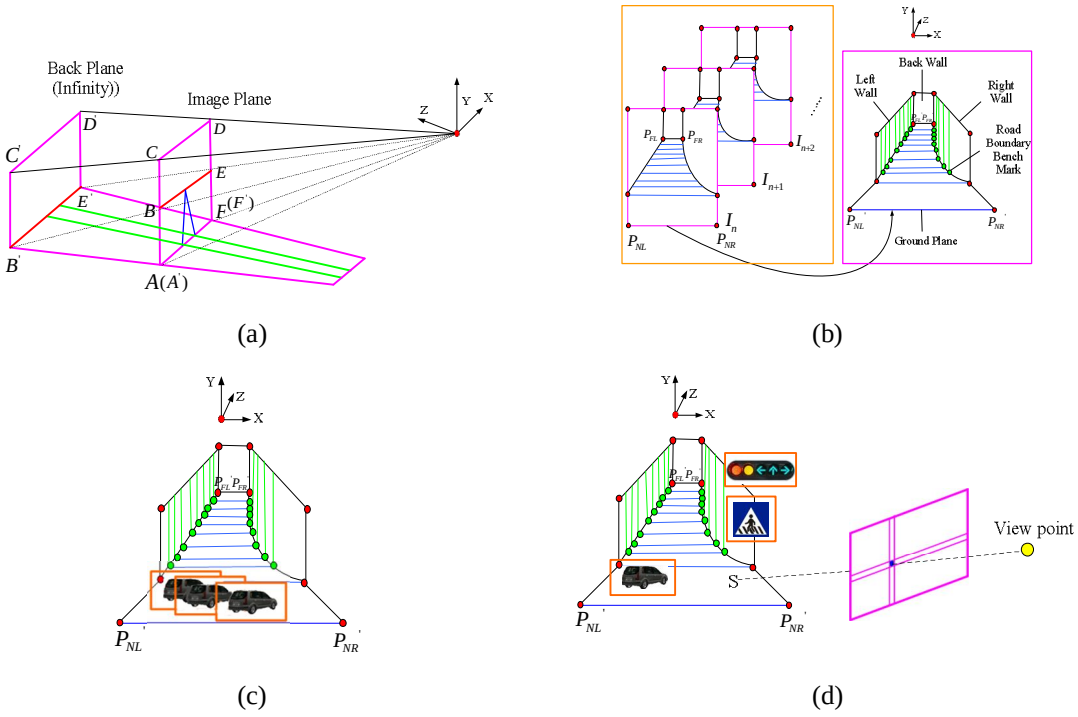


Figure 9: Background Modeling. (a) The scene model based on the vanishing line. (b) Our traffic scene model based on road boundary bench marks. (c) Foreground models construction. (d) new view point image rendering.

Foreground images and background images can be separated by previous sections. These images are further used to construct foreground models and background models respectively. We assume the camera is positioned at the origin, the view direction is toward $+z$, the view-up vector is toward $+y$, and the local length of the camera is f. Fig. 9 (a) shows the projection of scene model based on the vanishing line[24], where A, B, C, D, E, F are vertices in the image plane with focal length f. The space coordinates are extended into homogeneous coordinates: A', B', C', D', E', F'. The fourth element is 0 is the vertex is an ideal point., e.g. $B' : (x_B, y_B, f, 0)$, $F' : (x_F, y_F, f, 1)$.

Fig. 9 (b) shows our model composed of left wall, right wall, back wall and ground plane according to the control points. The control points can be propagated to each frame. P_{NL}, P_{FL}, P_{NR}, P_{FR} are projected to P_{NL}', P_{FL}', P_{NR}', P_{FR}', where P_{NL}', P_{FL}' are nearest

and farthest left control points while P_{NR}', P_{FR}' are nearest and farthest right control points respectively.

The homogeneous coordinates of these points are as follows:

$$P_{FL}' : (x_{FL}, y_{FL}, f, w_{FL}) \quad P_{FR}' : (x_{FR}, y_{FR}, f, w_{FR})$$
$$P_{NL}' : (x_{NL}, y_{NL}, f, 1) \quad P_{NR}' : (x_{NR}, y_{NR}, f, 1) \tag{22}$$

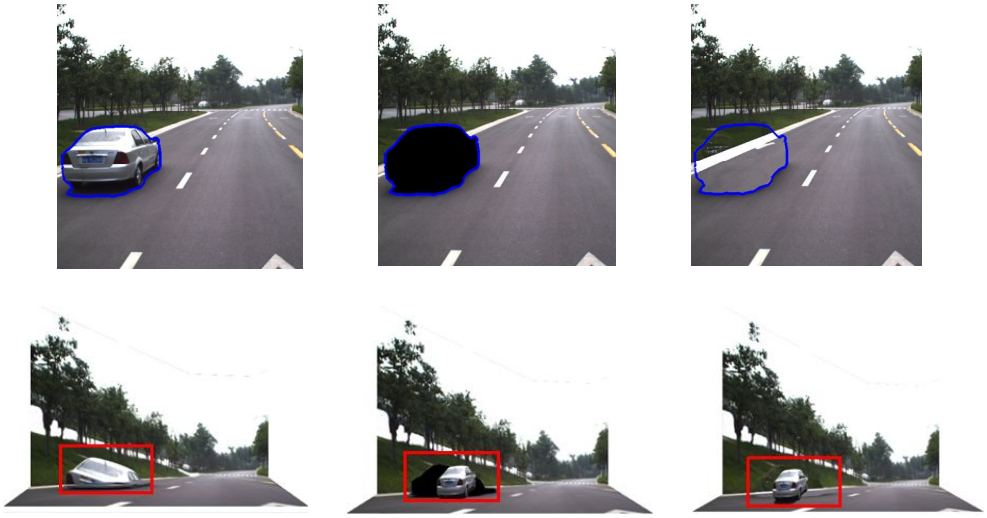
where w_{FL} and w_{FR} are fractional values. Vertices of other control points are computed respectively by:

$$w_{Li} = 1 - \frac{x_{Li} - x_{NL}}{x_{FL} - x_{NL}} + w_{FL} \bullet \frac{x_{Li} - x_{NL}}{x_{FL} - x_{NL}}, \quad w_{Ri} = 1 - \frac{x_{NR} - x_{Ri}}{x_{NR} - x_{FR}} + w_{FR} \bullet \frac{x_{NR} - x_{Ri}}{x_{NR} - x_{FR}} \tag{23}$$

Foreground objects are assumed to stand vertically on the ground plane. *RGBA* data structure is applied to these foreground objects, where *A* stands for the transparency ratio. Different objects have different transparency ratio, so we can have a hierarchical structure, as shown in Fig. 9 (c).

After the scene model is constructed, we can use the plenoptic function [25] to project the scene model onto the output image plane based on the view point (V_x, V_y, V_z):

$$PL = PL(\theta, \phi, \lambda, t, V_x, V_y, V_z) \tag{24}$$



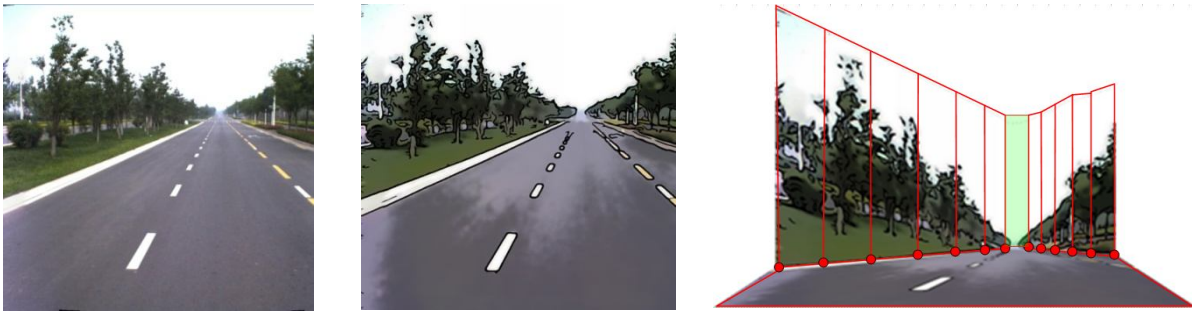
(a) (b) (c)

Figure 10: Foreground modeling. (a) without foreground modeling. (b) foreground modeling without background inpainting. (c) foreground modeling with background inpainting.

The constructed scene structure with foreground is shown in Fig. 10 (c). Fig. 10(a) shows the scene structure without foreground modeling. Fig.10 (b) shows the scene structure without background inpainting.

5.2 Non-photorealistic Rendering

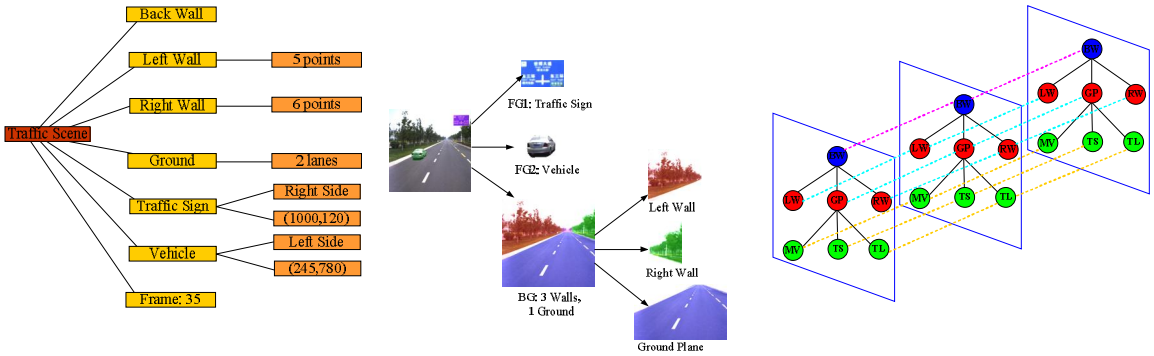
We applied non-photorealistic rendering to generate cartoon traffic images as a choice for users(Fig. 11). The cartoon images are produced by the methods described in [35][36].



(a) (b) (c)

Figure 11: Non-photorealistic scene structure. (a) original image. (b) cartoon image. (c) cartoon scene model.

5.3 Attachment relationships of traffic scene



(a) (b) (c)

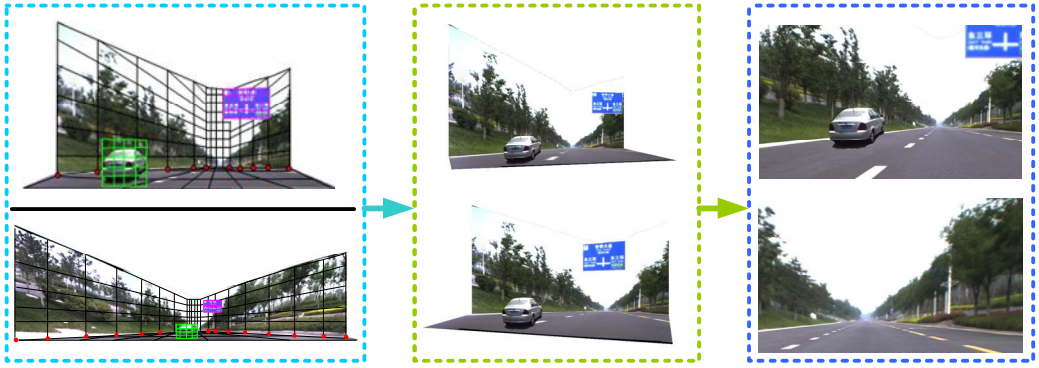
Figure 12: Attachment relationships of our scene model. (a) Attachement database for scene structure. (b) Hierarchical attachment relationships. (c) Graph model of scene elements corresponded to each frame.

A database which represents the attachment relationships of each traffic frame is constructed[37], as shown in Fig. The database records the quantity and positions of foregrounds for each frame, number of control points, etc.(Fig. 12 (a)). Fig. 12 (b) shows the hierarchical attachment relationships of our scene model. Fig. 12 (c) shows the graph model of traffic scene elements corresponded by each frame. The symbols have

following meanings: BW: Back Wall; LW: Left Wall; GP: Ground Plane; RW: Right Wall; MV: Moving Vehicles; TS: Traffic Signs; TL: Traffic Lights.

5.4 Touring experience based on the scene model

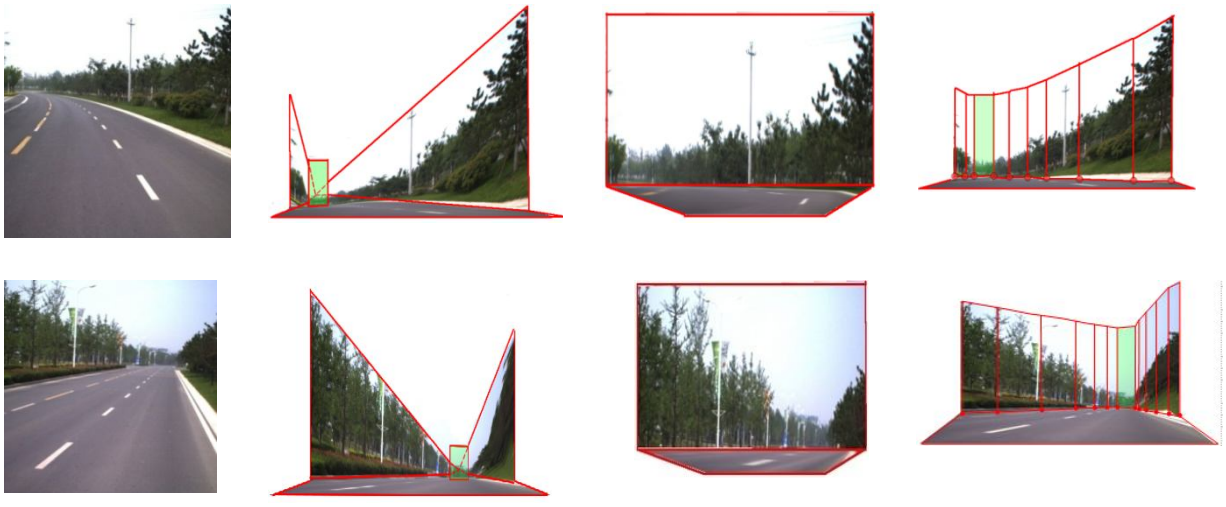
Fig. 13. shows our walkthrough images generation process. The scene structure can be changed by human-computer interaction. After the scene structure is specified, we can use steering wheel or joystick to tour into the scene models. The viewpoints and moving speed can be controlled by users.



(a) (b) (c)

Figure 13: Walkthrough images generation. (a) upper part: scene model based on single image lower part: scene model based on panoramic image. (b) scene model from different viewpoints. (c) walkthrough images generation.

6 Experiments



(a) (b) (c) (d)

Figure 14: Visual effect of different scene models. (a) original image. (b) TIP model. (c) TIP based on vanishing line. (d) our model.

Fig. 14 compares our scene model with TIP and TIP based on vanishing line. Our method can fit for curved road boundaries, thus is more suitable for traffic scene model construction.

Fig. 15 compares the functionality of our system with Microsoft Photosynth(MP), Microsoft Street Slide(MSS), Apple QuickTime VR(AQTVR), Google Street View(GSV) respectively. The diagram demonstrate that our system is more novel in that it has many properties: speed control, viewpoint chage, scene structure database, and non-photorealistic rendering.

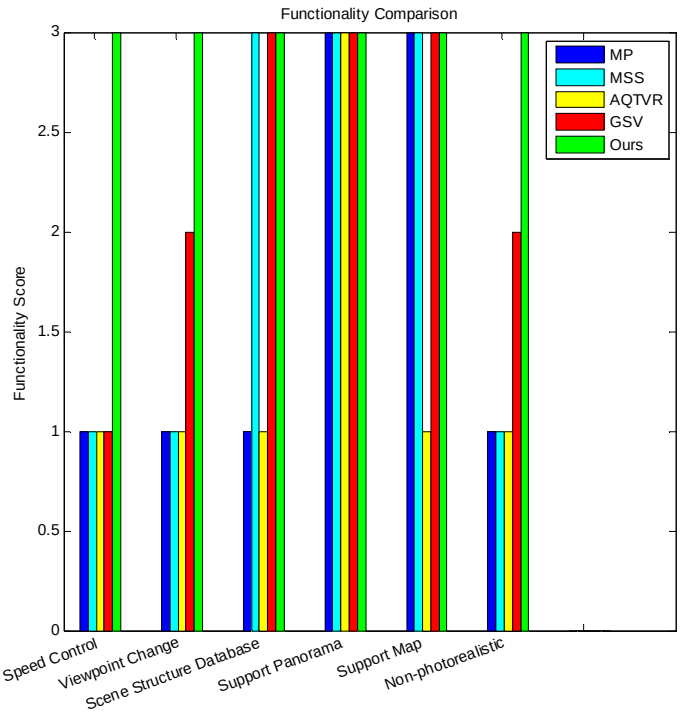


Figure 15: The functionality comparison. We compare our system with Microsoft Photosynth(MP), Microsoft Street Slide(MSS), Apple QuickTime VR(AQTVR), Google Street View(GSV). Different scores have following meanings: 1: no such function; 2: function in developing or partial fulfilled; 3: function complete.

We invited people of different ages to judge the novelty and usability of our system with MP, MSS and GSV. The ordinate axis (Fig. 16) indicates the percentage of the corresponding system's result, which perform best in all the 4 systems. The comparison score is based on different road types: city road, country road and freeway, with or without moving foregrounds. The diagram demonstrates our system performs best on city road and country road.

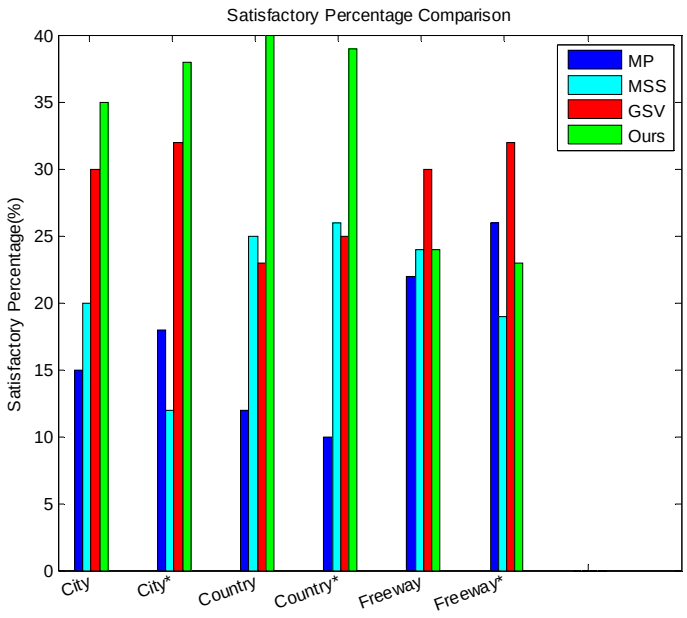


Figure 16: Satisfactory percentage comparison. The comparison score is based on city road, country road and freeway respectively. * means the road with moving foregrounds.

7 Conclusion and Future Work

In this work, we have propose 3D traffic scene models for users to tour freely, which are constructed by photos captured from cameras arranged on top of a moving car. We developed our own level set method to extract foreground objects. BP neutral networks, Gaussian process regression model and Gaussian mixture model are applied to inpaint the foreground regions by background information of adjacent frames. Our scene model is constructed according to road boundary control points, which can be propagated to each frame. Users can experience the feel of navigating into the traffic scenes after the scene models are constructed. Panoramic scene models and non-photorealistic scene models are also established to give users more free choice. We also developed our database to record scene model structure of each frame.

Our work is not meant to construct a full precise 3D scene model for every pixel of captured traffic pictures. The scene model is roughly composed of left wall, right wall, back wall, ground plane and foreground models. These are enough to give users convincing walkthrough experience. The experiments of foreground segmentation part demonstrate our superpixel based narrow band level set method is effective to extract foregrounds in traffic scene, and even more accurate if use a further MAP refine model.

For the scene model construction part, comprehensive user studies demonstrated the effectiveness of our system.

In the future, support vector machine (SVM) classifier will be applied to form the superpixel constellations to recognize road region and sky region for each frame. The recognized road region will help us to extract road boundary control points automatically. The sky region will be removed to enhance our scene model. We will recover the depth map for each scene structure based on the control points and further user annotations. The recovered depth map will help us to render 3D wireframe walls of scene structure.

ACKNOWLEGEMENTS

This work was supported by National Natural Science Foundation of China under Grant 91120009. The authors wish to thank Prof. Stanley Osher and the Mathematics Department of the University of California, Los Angeles for giving the visiting opportunity from December, 2012 to December, 2013.

References

[1] http://www.maps.google.com.

[2] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, "Google street view: capturing the world at street level," Computer, vol.43, No.6, pp. 32-37, 2010.

[3] J. Kopf, B. Chen, R. Szeliski, M. Cohen, "Street slide: browsing street level imagery," ACM Trans. on Graphics, vol.29, issue 4, pp. 102-106, July 2010.

[4] http://photosynth.net.

[5] E. Chen, "QuickTime VR- an image-based approach to virtual environment navigation," ACM SIGGRAPH, pp. 29-38, 1995.

[6] S.E. Chen, L. Williams, "View interpolation for image synthesis," SIGGRAPH 93, pp. 279-288, 1993.

[7] C.L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, R. Szeliski, "High-quality video view interpolation using a layered representation," ACM Transactions on Graphics, vol. 23, pp:600-608, 2004.

[8] A. Saxena, M. Sun, A.Y. Ng, "Make3D: learning 3D scene structure from a single still image," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 5, pp: 824-840, 2009.

[9] Y. Horry, K. Aanjyo, K. Arai, “Tour into the picture: using a spidery interface to make animation from a single image,” SIGGRAPH 97, pp. 225-232, 1997.

[10] H.W. Kang, S.Y. Shin, “Creating walk-through images from a video sequence of a dynamic scene,” Presence: teleoperators and virtual environments, vol. 13, No. 6, pp. 638-655, 2005.

[11] M. Tanimoto, “FTV: free-viewpoint television,” signal processing: image communication, vol. 27, no 6, pp. 555-570.

[12] S. Osher, J. Sethian, “Fronts propagation with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations,” J. Comput. Phys., vol. 79, pp. 12-49, 1988.

[13] J. Sethian, Level Set Methods and Fast Matching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science. Cambridge, U. K.: Cambridge Univ. Press, 1999.

[14] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, New York: Springer Verlag, 2002.

[15] T. Chan, L. Vese, “Active contours without edges,” IEEE Trans. Image Process., vol. 10, no. 2, pp. 266-277, 2001.

[16] D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang, “A PED-based fast local level set method,” J. Comput. Phys., vol. 155, pp. 410-438, 1999.

[17] F. Gibou, R. Fedkiw, “A fast hybrid k-means level set algorithm for segmentation,” in Proc. 4th Annu. Hawaii Int. Conf. Statistics and Mathematics, 2005, pp. 281-291.

[18] Y. Shi, W.C. Karl, “A real-time algorithm for the approximation of level-set-based curve evolution,” IEEE Trans. Image Process., vol. 17, no. 5, pp. 645-656, 2008.

[19] C.T. Hsu, Y.C. Tsan, “Mosaics of Video Sequences with Moving Objects,” Signal Processing: Image Communication, vol. 19, no. 1, pp. 81-98, 2004.

[20] Y. Wexler, E. Shechtman, M. Irani, “Space-time completion of video,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, pp. 463-476, 2007.

[21] J.F. Acker, B. Berkels, K. Bredies, M.S. Diallo, M. Droske, C.S. Garbe, M. Holschneider, J. Hron, C. Kondermann, M. Kulesh, “Inverse Problems and Parameter Identification in Image Processing,” Mathematical Metheods in Signal

Processing and Digital Image Analysis Understanding Complex Systems, pp. 111-151, 2008.

[22] K. Kim, D. Lee, I. Essa, “Detecting regions of interest in dynamic scenes with camera motions,” CVPR 2012, pp. 1258-1265.

[23] D. Hoiem, A.A. Efros, M. Hebert, “Automatic Photo Pop-up,” ACM Siggraph 2005.

[24] H.W. Kang, S.H. Pyo, K. Anjyo, S.Y. Shin, “Tour Into the Picture using a Vanishing Line and its Extension to Panoramic Images,” Computer Graphics Forum, vol. 20, no. 3, pp: 132-141, 2001.

[25] E.H. Adelson, J.R. Bergen, “The Plenoptic Function and the Elements of Early Vision,” Computational Models of Visual Processing, pp. 3-20, 1991.

[26] D. Mumford, J. Shah, “Optimal Approximation by Piecewise Smooth Functions and Associated Variational Problems,” Communication in Pure and Applied Mathematics, vol. 42, no. 1, pp. 577-685, 1989.

[27] G. Mori, “Guiding Model Search using Segmentation,” ICCV 2005.

[28] L.A. Zadeh, “Fuzzy Sets,” Information and Control. vol. 8, pp: 338-353, 1965.

[29] A.R. Mansouri, “Region tracking via level set PDEs without motion computation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 947-961, 2002.

[30] A. Bruhn, J. Weickert, C. Schn, “Lucas/Kanade meets Horn/Schunk: combing local and global optical flow methods,” International Journal of Computer Vision, vol. 61, no. 3, pp. 211-231, 2005.

[31] J. Mille, J.L. Rose, “Region tracking with narrow perception of background,” ICIP 2011.

[32] C. Rasmussen, C. Williams, “Gaussian Processes for Machine Learning,” MIT Press. 2006.

[33] Rumelhart, E. David, Hinton, E. Geoffrey, Williams, J. Ronald, “Learning representations by back-propagating errors,” Nature, vol. 323, pp: 533-536, 1986.

[35] A. Gooch, B. Gooch, P. Shirley, E. Cohen, “A non-photorealistic lighting model for automatic technical illustration,” ACM Siggraph 1998.

[36] P. Litwinowicz, “Processing Images and Video for An Impressionist Effect,” Proceedings of the 24th annual conference on computer graphics and interactive techniques, pp. 407-414, 1997.

[37] B.C. Russell, A. Torralba, “Building a database of 3D scenes from user annotations,” CVPR 2009.

[38] S.C. Zhu, A. Yuille, “Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 9, pp: 884-899, 1996.

[39] P.E. Debeve, C.J. Taylor, J. Malik, “Modeling and rendering architecture from photographs: a hybrid geometry-and image-based approach,” ACM Siggraph 1996.

[40] S. Wang, H. Lu, F. Yang, M.H. Yang, “Superpixel tracking,” ICCV 2011.

[41] S. Gobron, G. Marx, J. Ahn, D. Thalmann, “Real-time textured volume reconstruction using virtual and real video cameras,” Computer Graphics International, 2010.

[42] F. Bornemann, T. Marz, “Fast image inpainting based on coherence transport,” Journal of Mathematical Imaging and Vision, vol. 28, no.3, PP: 259-278, 2007.