

A High-Resolution Surface Image Capture and Mapping System for Public Roads

Yazhe Hu and Tomonari Furukawa
 Virginia Tech

ABSTRACT

This paper presents a system designed to develop a high-resolution map of public roads by capturing high-resolution surface images. Unlike conventional system, the proposed system applies a field programmable gate array (FPGA) to synchronize camera, Inertial Measurement Unit (IMU), and Global Positioning System (GPS) by using FPGA's high clock frequency and flexibility to multiple devices. The proposed system, which can be mounted on a regular vehicle, contains a Complementary Metal–Oxide–Semiconductor (CMOS) camera which can achieve 0.006 ms shutter speed and 150 fps frame rate. This camera's high shutter speed and high frame rate can help capturing images with overlapping region at fast driving speed so that no area is missing from road surface image capturing. The images collected by the proposed system contain clear patterns without motion blur, and feature points on images and feature matching can be found by using existing algorithm like Scale-invariant feature transform (SIFT) between any two consecutive images. Camera's vertical facing towards road surface enables using affine transformation to reduce computational time of registration by calculating only six instead of eight parameters using homography matrix. IMU and GPS data are synchronized and recorded together with image capturing, and is applied to reduce accumulated error for long-distance mapping by providing location information of vehicle for image registration process. A clear map stitched by millimeter-resolution images is obtained at up to 100 km/h and is registered and uploaded to cloud. The road surface mapping can be used in road survey or road maintenance and can visualize public road surface condition. In this paper, the effectiveness of this system and mapping method is illustrated qualitatively and quantitatively through experimental evaluation.

CITATION: Hu, Y. and Furukawa, T., "A High-Resolution Surface Image Capture and Mapping System for Public Roads," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 10(2):2017, doi:10.4271/2017-01-0082.

INTRODUCTION

Road surface monitoring is an important work for public transportation system. The maintenance of a healthy road surface condition can improve the safety for public traffic and result in a longer life for roads. There is a lot of money spent every year for road maintenance. Since most of the road surface survey works are still done by human workers, the cost of human labors keeps increasing and it is not only unsafe but also inefficient for workers to search and detect road defects positions on public roads. On the other hand, there is a need for cars to get more information on the road surface in order to recognize road markings and avoid road defects as well. In this case, a high-resolution clear road surface image capturing and mapping system for public road is needed.

Existing road surface capturing systems can be categorized into two parts: the low-speed road surface capturing system and the high-speed road surface capturing system. The low-speed system means the system could capture road images statically or at a low speed which was far behind regular public traffic driving speed. Ahmed et al. [1] proposed a system which used only one CMOS camera and

reconstructed 3-dimensional (3D) road surface by photogrammetric technique. He used a single camera and statically captures road surface images in different angles. Then by manually labeling the feature marks a scaled 3D road surface can be reconstructed. Antol et al. [2] deployed a structured-light based RGB-D camera to capture road surface profile. In this system, a movable box was applied as a cover to avoid sunshine and could move with a low speed at outdoor environment. However, although these two systems could get 3D information for road surface, they were static or low-speed systems which were not suitable for image capturing on public road where traffic is moving fast.

High-speed road surface image capturing system could have a driving speed which satisfied the public road condition. Some of the systems were using line-scan cameras to capture road surface images. AMAC [3] system applied a line scan camera with laser illuminator, and can capture road with 4m width at 80 km/h driving speed. MPSV [4] used line-scan camera with 2mm resolution and has a shutter speed up to 1/40000 s to limit the motion blur. LRIS [5] system mounted two line-scan camera to enlarge camera field of view, and it can get 9096 × 2048 pixels image after collecting many line scanings. LRIS could

achieve 100 km/h driving speed. These line-scan camera systems could operate at high speed but they only got a one-dimensional (1D) line image at each scanning which could not provide two-dimensional (2D) images. Some other systems applied video cameras to collect road surface images. ROMDAS [6] used video cameras combined with Global Positioning System (GPS) and Inertial Measurement Unit (IMU) to capture road images at frame rate of 2 fps and a vehicle speed of 50 km/h. MMS [7] and a system proposed by Miraliakbari et al. [8] used a video camera plus a laser scanner to capture 3D road surface images. However, these systems using video cameras either had low camera frame rate or low shutter speed so that they cannot get clear road surface images without motion blur at a highway speed (more than 100 km/h).

This paper presents a system which can capture high-resolution road surface images for public road at highway speed. This system is composed of a field programmable gate array (FPGA), a Complementary Metal–Oxide–Semiconductor (CMOS) camera, IMU, GPS, and a computer. The FPGA can provide a high frequency synchronizing signal to all the devices. The CMOS camera used in this system has high frame rate which can capture consecutive images with overlapping region, and high shutter speed which can significantly reduce image motion blur. IMU and GPS provide camera pose information which can help speeding up the road image stitching and registering road map to the cloud.

This paper is organized as follows. The following section introduces problem formulation and reviews image feature finding and matching, and image stitching. The third section describes the proposed road surface image map stitching and the road surface image capturing system design. Then next section shows the road map stitching result and the registration of the road surface map to Google Map. Conclusions are summarized at the last section.

ROAD SURFACE IMAGE CAPTURING

Figure 1 shows the use of one camera to capture road surface images. The camera is mounted vertically to the road surface and facing towards road as shown in the figure. The $[x, y, z]^T$ represents the points in camera image sensor coordinate, while $[u, v]^T$ and $[X, Y, Z]^T$ represents points in image coordinate and world coordinate separately. The camera first takes one image at position 1 and moves to position 2 to take another image. These two images contain overlapping regions, which means some of the parts in these images represent same area of road surface. The camera then keeps moving forward and meanwhile capturing n images $\{A, B, C, \dots, N\}$ for road surface. Every two consecutive images should have overlapping region the same as Image 1 and Image 2 showed in Figure 1. Feature matching and image stitching techniques are then used to stitch images together as a road map. In this section, feature detecting and matching will be discussed first, and then image stitching will be explained by using corresponding points from feature matching.

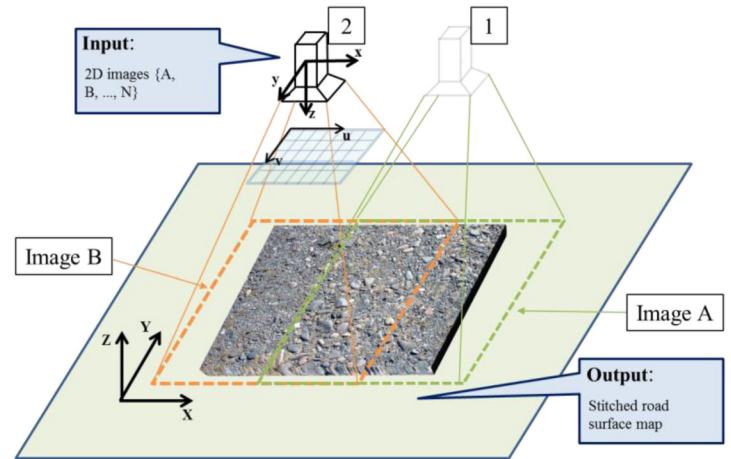


Figure 1. Image capturing on road surface and coordinate systems for camera sensor, image, and world.

A. Feature Finding and Matching

In order to stitch the overlapping regions together between two consecutive images, features on the images should be described and same features between two consecutive images need to be matched. Scale Invariant Feature Transform (SIFT) feature [9] is a scale-invariant and orientation-invariant feature descriptor which can be used to detect features on road surface images. There are three steps for SIFT to find and match features: feature detection, feature description, and feature Matching.

1. Feature Detection

Difference of Gaussian (DoG) function is used to find the keypoint locations and the scale of features. Equation 1 shows the Gaussian kernel while Equation 2 illustrates the difference of Gaussian equation:

$$G(u, v, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(u+v^2)/2\sigma^2} \quad (1)$$

$$\begin{aligned} D(u, v, \sigma) &= (G(u, v, k\sigma) - G(u, v, \sigma)) * I(u, v) \\ &= L(u, v, k\sigma) - L(u, v, \sigma) \end{aligned} \quad (2)$$

Where σ is the scale parameter, (u, v) is the coordinate of points on image, $*$ is the convolution operator, and L is a function that represents the convolution of Gaussian kernel with image $I(u, v)$. The feature location can then be determined by finding local extrema of DoG, while the scale parameter σ can determine the scale of features. Local extrema includes local maxima and local minima, and is determined by checking the DoG values of a pixel's neighboring DoG values.

After getting the location of the features, orientation of features are also computed to make these features rotation-free. The gradient magnitude and orientation of each pixel in the image are illustrated in Equation 3 and Equation 4:

$$m(u, v) = \sqrt{(L(u+1, v) - L(u-1, v))^2 + (L(u, v+1) - L(u, v-1))^2} \quad (3)$$

$$\theta(u, v) = \tan^{-1}((L(u, v+1) - L(u, v-1)) / (L(u+1, v) - L(u-1, v))) \quad (4)$$

within a region around the feature point, m and θ are calculated for each pixel inside this region and a histogram of θ is obtained from calculation. The orientation of a feature point can be defined by the θ which is maximum within orientation histogram.

2. Feature Description

After getting the location, scale, and orientation of features, a descriptor is used to describe every feature. Figure 2 shows a descriptor with 4×4 subregions which using histogram to describe a feature. For each subregion, gradient m and orientation θ are calculated for every pixel inside this region. Then all of the m and θ are analyzed using a histogram which contains 8 bins to represent 8 directions. Thus for all of these 4×4 subregions, there are $4 \times 4 \times 8 = 128$ elements to describe a feature. So far every feature point contains its location, scale, orientation, and a 128-element vector to describe itself.

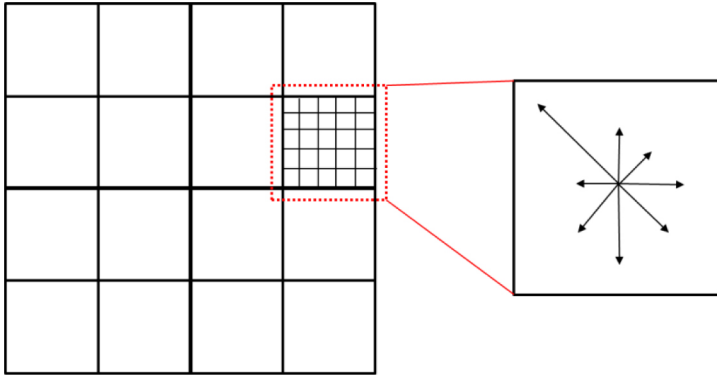


Figure 2. feature descriptor with 4×4 subregions using gradient magnitude m and orientation θ .

3. Feature Matching

After getting the location, scale, orientation, and a 128-elements descriptor of every feature, matching should be deployed to find the corresponding features between two consecutive images. Figure 3 demonstrates the feature matching between two consecutive images with overlapping region. Equation 5 is used to match corresponding features between two images:

$$[m_i, m'_i] = \{[u_i, u'_j] \mid \arg \min_{u'_j} (D(u_i, u'_j))\} \quad (5)$$

where m_i is the position of i th matched feature in the first image, m'_i is the position of i th matched feature in the second image. u_i is the location (u, v) of i th feature in the first image, u'_j is the location (u', v') of j th feature in the second image, and $D()$ is a function of Euclidean

distance calculated by using the two 128-elements descriptors at the two features located at u_i and u'_j separately in two images. Two features are considered as corresponding features if the Euclidean distance between them is the smallest among all the feature pairs.

B. Image Stitching

Image stitching is the process of using 2-dimensional (2D) images and stitching them in a same 2D coordinate system. There are mainly two conditions in 2D image stitching, one is the images preparing to be stitched are representing object in different plane, another is the images waiting to be stitched represent object in the same plane. If the camera is mounted vertically facing towards the road surface, the image stitching problem becomes stitching objects in the same plane. This simplification will reduce necessary number of corresponding points used in stitching images together, and this simplified image transformation is called affine transformation. In affine transformation, parallel lines are remaining parallel after it is transformed to new coordinate.

Figure 3 shows the two neighboring images A and B with same features in the overlapping region. In this case, 2D affine transformation can be used to get the transformation parameters between two neighboring images. Equation 6 shows the affine transformation in this case, where $[u', v']^T$ is the point in image A and $[u, v]^T$ is the point in image B.

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} u' \\ v' \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} \quad (6)$$

Since the transformation parameters m_1, m_2, m_3, m_4, t_1 , and t_2 are the needed outputs, Equation 6 can also be rewritten to be Equation 7.

Equation 8 applies a variable A and b to represent the $n \times 6$ matrix and a $n \times 1$ vector accordingly in Equation 7.

$$\begin{bmatrix} \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ u'_i & v'_i & 0 & 0 & 1 & 0 \\ 0 & 0 & u'_i & v'_i & 0 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \dots \\ \dots \\ u_i \\ v_i \\ \dots \\ \dots \end{bmatrix} \quad (7)$$

$$Ax = b \quad (8)$$

Equation 9 is obtained by solving x in Equation 8 and x contains all the transformation parameters:

$$x = (A^T A)^{-1} A^T b \quad (9)$$

Finally, after getting those transformation parameters between two images, image A and image B can be transformed to a same coordinate system and stitched together.