Lane detection using Fourier based line detector

Konstantin Burlachenko
bruziuz@stanford.edu

Abstract

Traffic accidents account for a vast majority of fatalities worldwide. The Ministry of Public Safety of China reported 667 507 of traffic-related accidents in 2003 as fatal. [13] It has always been a goal for engineers to get to the point of accident free driving. Developing a full autonomous navigation system to assist the driver by alerting him/her in hazardous situations or even take part of the driving task has been an active field of car accident research for the past two decades. Driver assistance systems help drivers to better prepare in unpredictable instances and react instantly when necessary. All of the existing system perception problems can be categorized into two groups: namely, lane/road detection and obstacle detection." [1]

As a final project I have decided to replicate paper [1]. In this paper a new approach to lane marker detection problem is introduced. This paper mostly use this previous works: [9],[13],[14], [15],[16],[17],[20]. My work contains a full description of method without a lot of cross references to part of algorithms as it was done in [1]. I tried to do this work as complete and self-contained as possible. The presented method in [1] incorporates advance line detection using multilayer fractional Fourier transform (MLFRFT) for Radon Transform and the state-of-the-art advance lane detector (ALD). Radon Transform is a technic from medical imaging and signal processing. The problem of Tomography is to reconstruct 2d image from knowledge of how light (X-rays) intensity decreased in 1D sections. This problem has been solved via Radon Transform. But also this transform can be used for line detection problem. Radon transform has received much attention for its efficiency and accuracy compared with Hough transform [20]. Radon Transform has some properties which make him looks like Hough Transform, but it is not the Hough transform.

1. Introduction

As a final project I have decided to replicate paper *[1] Payam S. Rahmdel, Daming Shi, and Richard Comley: "Lane detection using Fourier-based line detector"*. Humans do decision making during be a driver mostly based on their vision.

Lane marker detection and tracking it is still a fertile area of machine vision research. The main problems in Lane detection are: (1) The presence of shadows, producing artifacts onto the road surface.(2) The presence of other vehicles on the path, partly occluding the visibility of the road. (3) Some systems should be designed to work on unstructured roads. The lane detection problem is to detect and track the driving lane boundaries and estimates the geometry of the lane in the 3D world coordinates relative to the vehicle. Society need autonomous lane detection due to progress in self driving cars and automobile industry. In this work no special hardware devices s.t. monocular, stereo cameras, lidar, inertial measurement unit (*imu*), global position system (gps), radar are used. Instead it in this work only images from calibrated pinhole camera are used (Intrinsic and extrinsic parameters).

My interest in this work is due that have been based on Fourier Transform. In class EE261 in Stanford - The Fourier Transform and Its Applications prof. Brad Osgood in 2015 started a class with words: "Methods based on the Fourier transform are used in virtually all areas of engineering and science and by virtually all engineers and scientists." At the epigraph of the book [10] there are such words: As Lord Kelvin said, "Fourier's theorem is not only one of the most beautiful results of modern analysis, but it may be said to furnish an indispensable instrument in the treatment of nearly every recondite question in modern physics.".

So when I started to work at this problem I hoped that methods based on Fourier Transforms (e.g. Hartley transform, Cosine transform, Radon Transform) can be used to leverage in solution of the problem with lane detection.

2. Review of previous works

Lane detection is a crucial component of most computer vision based driver assistance systems. It needs to be extremely reliable and robust. Lane detection is a well studied research problem. Numerous techniques have been developed to detect and extract lanes in an automobile using a camera.[9].

Work [1] "Lane detection using Fourier-based line detector" is based on [13] "A Novel Lane Detection System With Efficient Ground Truth Generation". Work [13] was based on [9] "A Layered Approach To Robust

Lane Detection At Night". Which have original idea of specific lane detection architecture.

So the chain of evolution is the following: [9],[13],[1]. Each work append something new. Before work [9] the following approaches have been used:

1.Canny edge detector (only) approach: Find edges via Canny edge detector and the use Hough Transform to find the best fitting pair of lines.
Canny edge detector (only) problems: Artifacts on the road, such as navigational text, arrows, etc. will often have features that appear to be straight lines in the edge detected images. In these cases, the Hough transform may incorrectly classify this edges as lane markers.

2. Use color cameras approach: Use color RGB channels to convert into HSV or HSL model to distinguish lane markers from other things in the road.
Use color cameras problems: Problems with color segmentation arise when the illumination has a non-white color such as yellow or amber that is found in most street lights. Intensity values are unlikely to remain consistent over the different construction materials used to build the road.

3. Use GPS approach: Use GPS with digital maps to determine the position of the vehicle with respect to the road and extrapolate the lane marker locations.
Use GPS problem: GPS enabled devices are restricted to operation under an open sky, and require having up to date digital maps. GPS device is also prone to errors of up to a couple of feet thereby making their use in lane detection of questionable utility.

The work [9] approach is based on image enhancement filtering, low resolution Hough transform. One of the interesting thing which was introduced in [9] and then is used in [13] is the adaptive local threshold. Authors of [9],[13] said that this approach tends to work better than the popular Canny edge detector, which uses an adaptive global threshold to convert the gradient edge image to binary [13]. The global threshold may prohibit the detection of weak edges. The work [13] append to [9] Inverse Perspective mapping for image transformation, RANSAC for outlier removal. Difference between [1] and [13] will be discussed in next sections.

3.1 Summary of the technical solution and why it is better then previous work

Technical solution from works [9], [13] is foundation for [1] and can be described in the following table.

Name	The purpose of usage
1 Region of interest selection (ROI)	Portion of image that has the higher probability of encompassing the lane markers. ROI is usually selected between the vanishing point and the vehicle's hood
2 Temporal blurring	Temporal blurring connects the broken lane markers by producing an average image of current frame and a number of predecessor frames
3 Bird's eye view	Inverse perspective mapping (IPM) transforms image from camera perspective view to bird's eye view orthogonal view. In camera view, the width of lane markers changes depending on its distance. Parallel lines are non-parallel in perspective camera frame, but bird's eye view provides parallel line separated in specific distance
4 Edge detection	Because the HT is only applicable on binary images, some mechanism should exist to obtain a binary image. Adaptive thresholding algorithms have been proposed in [13],[9]
5 Standard Hough transform	The image is split into halves and HT will be applied to each half separately. Quality of this step is depend on previous step 4.
6 Peak detection	The HT peak detection determines the potential candidate for the lane marker
7 Sampling detected lines	Post-process. Each candidate line has to be sampled by a predefined resolution
8 1D template matching	Post-process. At each sample point, 1D template matching will be applied to estimate the best matching pixel
9 Least square estimation	RANSAC to eliminate ouliers. And minimize the error and locate the best matching straight line, a linear least square estimation (LSE) is applied

[Table 1]

New in technical solution from [1] is that the old steps 4,5,6 from previous section have been removed. It is so because proposed method in [1] is based on a Multi Layer Fractional Fourier transform (MLFFT) which can accept the monochrome data as its input. Therefore, prior binarization via step#4 and edge detection via #5, #6 is not required. So this steps #4,#5,#6 have been replaced by new step:

4-6: MLFFT	

MLFFT is a technic to reconstruct continuous Radon Transform more precisely. Radon Transform for intensity function *f(x,y)* in theory is a continuous function $R_f(L)$ which give curved integral of *f(x,y)* over line L. For details please look at *"Advanced Radon Transform with MRFLT for straight line detection"* section.

MLFFT and Radon approach it is better to edge detection because:

1) Hough Transform take input is binary image. Radon Transform input is intensity of the pixels *f(x,y)* in the usual Cartesian grid. This method is

independent of previous edge detection. And different thresholds relative to select edges.

2) Grid size parameter for Hough Voting is rather hard to find hyper-parameter. For Radon Transform and MLFFT polar grid high quantization doesn't have any downsides. It's due there are no explicitly voting procedure by cells.

3) Because Radon Transform pipeline incorporate Fourier Transform it is possible to concentrate on some things in the frequency grid. E.g. for the edges it's more about high frequencies.

4) Also due to working in frequency spectrum it's possible to apply some filters prior to 1D inverse Fourier Transform during working with Slice Theorem.

5) Radon transform is interesting transform in terms special convolution, shifting, rotation, scaling theorems for this conversion

Radon Transform output is still in the form of sinogram like in Hough Transform. But still some method about peak selection should be applied.

3.2 Technical part

3.2.1 Region of interest selection

An ROI is extracted by averaging color channels [9]:

$$I(x,y) = \frac{R(x,y) + G(x,y) + B(x,y)}{3}$$

The purpose of this stage is possible to reduce the computational complexity of the algorithm by initializing a useful region of interest (ROI) of lane detection [6].

3.2.2 Temporal blurring

Traffic lane markers on highways are dashed lines and, depending on the exposure time, they may appear as a dot or a short line in the image. Temporal blurring may be used to extend the lane markers and give the appearance of a long and continuous line. The temporal component of the blurred image is only a few milliseconds to avoid ghosting due to motion blur. On curving roads, the curvature of the road changes gradually as opposed to being abrupt. Therefore, lane markers are only slightly affected at the entry and exit points of the curve. In addition, the blurring technique helps to reduce noise in the image assuming the presence of zero mean Gaussian noise. Authors of [1] didn't mention which technic they used, but In [13] it was mentioned this formula: $AverageImage = \sum_{i=1}^{N} \frac{I_{current-i}}{N}$

3.2.3 Bird's eye view

Bird's eye view or Inverse Perspective mapping is used to transform the captured images from camera perspective to a bird's eye view. Due to the perspective effect the road lane width changes according to their distance from the camera. With IPM:

- Lane detection now becomes a problem of locating a pair of near-constant-width parallel lines
- Enables a mapping between pixels in the image plane to a world coordinates
- Each pixel represents the same portion of the road

To remove the perspective effect, it is necessary to know camera extrinsic and intrinsic parameters and the scene properties represented in the image (e.g. the road, which is now assumed to be flat). It is so because all points from Euclidian space in the ray between "Center of the camera" and "Element of retina" in pinhole camera model will be projected into this sensitive element.

So to construct IPM extra assumptions are needed. Unfortunately there are no information in article [1],[13] how authors implement IPM, only one picture which doesn't talk anything. I have found rather specific description in [19]:

$$x^* = X_0 - \frac{Z_0 x_p \cos(\theta) + \left(y_p \sin(\varphi) - f\cos(\varphi)\right) Z_0 \sin(\theta)}{y_p \cos(\varphi) + f\sin(\varphi)}$$

$$y^* = Y_0 - \frac{Z_0 x_p \sin(\theta) - \left(y_p \sin(\varphi) - f\cos(\varphi)\right) Z_0 \cos(\theta)}{y_p \cos(\varphi) + f\sin(\varphi)}$$

(Which belongs to Duda & Hart, 1973)
It's close solution, where:
$(x^*, y^*, 0)$ – ground point coordinates in w.c.s.
(X_0, Y_0, Z_0) – coordinate of the camera in w.c.s. in moment when the frame has been taken
(θ, φ) – camera yaw and pitch angles

But in "Duda & Hart" approach it was assumed that pixels are square, no offset, no skewnes.

I derived more general solution. Assume that we have some general K,R,T matrices for camera pinhole model without distortion:

$$\begin{aligned}\lambda[x_p\ y_p\ 1]^T &= K_{3x3}[R\ T]_{3x4}[x^*\ y^*\ 0\ 1]^T \\ &= M_{3x4}[x^*\ y^*\ z^*\ 1]^T \\ &= \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix}[x^*\ y^*\ z^*\ 1]^T\end{aligned}$$

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} \frac{m_1 P^*}{m_3 P^*} \\ \frac{m_2 P^*}{m_3 P^*} \end{bmatrix} \Rightarrow \begin{matrix} x_p(m_3 P^*) - m_1 P^* = 0 \\ y_p(m_3 P^*) - m_2 P^* = 0 \end{matrix} \Rightarrow$$

$$\begin{cases} x_p(m_{31}x^* + m_{32}y^* + m_{33}z^* + m_{34} \cdot 1) - (m_{11}x^* + m_{12}y^* + m_{13}z^* + m_{14} \cdot 1) = 0 \\ y_p(m_{31}x^* + m_{32}y^* + m_{33}z^* + m_{34} \cdot 1) - (m_{21}x^* + m_{22}y^* + m_{23}z^* + m_{24} \cdot 1) = 0 \end{cases}$$

$$\begin{cases} (x_p m_{31} - m_{11})x^* + (x_p m_{32} - m_{12})y^* + (x_p m_{33} - m_{13})z^* + x_p m_{34} - m_{14} = 0 \\ (y_p m_{31} - m_{21})x^* + (y_p m_{32} - m_{22})y^* + (y_p m_{33} - m_{23})z^* + y_p m_{34} - m_{24} = 0 \end{cases}$$

$$\begin{cases} (x_p m_{31} - m_{11})x^* + (x_p m_{32} - m_{12})y^* + (x_p m_{33} - m_{13})z^* = m_{14} - x_p m_{34} \\ (y_p m_{31} - m_{21})x^* + (y_p m_{32} - m_{22})y^* + (y_p m_{33} - m_{23})z^* = m_{24} - y_p m_{34} \end{cases}$$

This system has Rank 2 or 1(in degenerate case), but it is exist three unknowns. Right now we can use extra assumption about ground plane $z^* = 0$ and solve this system with respect to x^*, y^*. For some test datasets (e.g. KITTI benchmark) ground it was assumed that ground plane is $y^* = 0$.

3.2.4 MRLFT stage

3.2.4.1 Radon Transform and projection slice theorem

In medical imaging there is exist interesting transformation used for tomography, and similar mathematics treatment used for magnetic resonance imaging (MRI). Key transformation for tomography is Radon transform. Please, look into 7.1 section for details.

Original definition of the Radon transform is the curved integral in the line L of the function $f: \mathbb{R}^2 \to \mathbb{R}$. Ln coordinates it can be written vias using line delta function.

$$R_f(L) = R_f(\theta, \rho) = \int_L f = \iint_{\mathbb{R}^2} f(x_1, x_2)\delta_L dx_1 dx_2$$
$$= \iint_{\mathbb{R}^2} f(x_1, x_2)\delta(\rho - x_1 cos(\theta) - x_2 sin(\theta))dx_1 dx_2$$

Radon Transform has interconnection with 2D Fourier Transform, and 1D Fourier Transform.

$$(\mathcal{F}(R_f, and\ fix\ \theta))(r, \theta)$$
$$= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x_1, x_2)\, dx_1 dx_2 \left(e^{-2\pi i(x_1\zeta_1 + x_2\zeta_2)}\right)$$
$$= \mathcal{F}_{2d} f(\zeta_1, \zeta_2)$$

Where:

$$\zeta_1 = rcos(\theta),\ \zeta_2 = rsin(\theta)\ ,$$
$$\zeta_1{}^2 + \zeta_2{}^2 = r^2,\ \ tan(\theta) = \frac{\zeta_2}{\zeta_1}$$

This means that via 1d Fourier transform w.r.t ρ and fixed θ we know 2d Fourier Transform of f in ζ_1, ζ_2 points. Process of reconstructing f from $\mathcal{F}_{2d} f$ is called ***Projection Slice Theorem or Central Slice Theorem***. The problem with this theoretical transformation is that $(\mathcal{F}(R_f, and\ fix\ \theta))(r, \theta)$ defined in polar coordinates But inverse Fourier Transform of $\mathcal{F}_{2d} f(\zeta_1, \zeta_2)$ should be done in Cartesian grid.

3.2.4.2 Reconstruct Radon Transform via projection slice theorem

Due to Fourier Slice Theorem, instead of calculating line integrals in the spatial domain it is feasible to do:

1) Transfer image into frequency domain via 2D FT of the image
2) Interpolate the Cartesian grid to Polar
3) Fix θ along a radial line at this angle interpolate values from (2) to create sample function
4) For fixed θ make 1D inverse discrete FT
5) Repeat 3-4 steps and receive $R_f(\theta, \rho)$

Authors suggest to use DFT to approximate FT, and IDFT to approximate IFT. DFT transformation is approximation of Fourier Transform up to the scaling factor, pleas look into 7.2 section for details.

3.2.4.3 MLFFT and MLFFT grid

Theory of Multilayer Fractional Fourier Transform was introduced in [16]. In this work this technic was used for *Image Registration* problem. This problem is about transform an image to align its pixels with those in another image. Map the coordinate (x, y) of an image to a new coordinate (x', y'). Transformation can be linear or nonlinear.

In [16] the following properties of MLFFT for image have been observed:

1. High accuracy in recovering large scale factors and large rotation angles
2. Adaptability for different precision requirements
3. Working well with both the log-polar and the polar transforms

DFT transform was defined like this:

$$Ff[n] = \sum_{k=0}^{N-1} f[k]e^{-\frac{2\pi i n k}{N}}$$

Fractional Fourier transform was defined with extra "a" parameter:

$$F^a[k] = \sum_{n=0}^{N-1} f[n]e^{-\frac{2\pi i k}{N}an}$$

Centered version of Fractional Fourier Transform is:

$$F^a[k] = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} f[n]e^{-\frac{2\pi i k}{N}an}, -\frac{N}{2} \le k \le \frac{N}{2} - 1$$

When a=1 => we get N frequencies uniformly distributed with spacing equal to 1/L between each other. For DFT the spacing in one domain is determined by properties of the function in the other domain. Look into 7.2 section for more details.

2d Fractional Fourier transform was defined similar:

$$F^a[k_1, k_2] = \sum_{n_1=-\frac{N}{2}}^{\frac{N}{2}-1}\sum_{n_2=-\frac{N}{2}}^{\frac{N}{2}-1} f[n_1, n_2]e^{-\frac{2\pi i n_1}{N}ak_1}e^{-\frac{2\pi i n_2}{N}ak_2}$$

Theorem about Multi Layer Fractional Fourier: Transform (MLFFT), derived in [16], page 4:
For $0 \le \alpha \le 1$ $F^a[u, v] = F[au, av]$

2d Fractional Fourier transform gives points which lie in integer and in not integer grid. Let's now introduce some concepts before introduce MLFFT grid.

Number of approaching levels is defined as: $L \geq 1$
The approximation cut is defined as:

$Cut = \{0 < a_i < a_{i+1} < \cdots a_L = 1\}$

MLFFT grid in frequency domain:

$$P_i = \left\{(a_i u, a_i v), -\frac{N}{2} \leq u, v \leq \frac{N}{2} - 1\right\}, a_i \in Cut$$

Each point P_i corresponds to a certain frequency on the 2D frequency domain plane. Suppose we have image [N+1, N+1] then by applying the 2D fractional Fourier Transform we can evaluate the value of any frequency at P_i. And finally we union all point into final grid $P = \cup_{i=1}^{i=L} P_i$

Example:

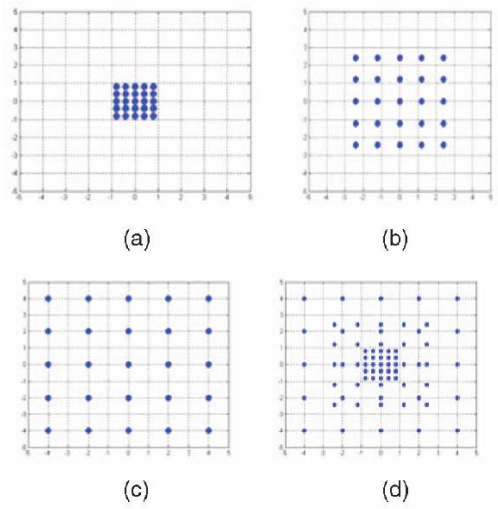

Fig. 1. The construction of an MLFFT grid with $\{Cut\} = \{0.1,($ (a) P_1 grid. (b) P_2 grid. (c) P_3 grid. (d) P grid.

MLFFT grid:

1) Perform better than other methods for the polar FT
2) Also better than other methods for the log-polar FT
3) The resulting frequency response of the image has more frequency samples compared with traditional one-layer 2-D DFT.

MRLFT can be more enhanced via Generalized interpolated Fourier Transform (GIFT). GIFT method was mentioned in [14],[20]

$$F^{a,b}[k_1, k_2] = \sum_{n_1=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{n_2=-\frac{N}{2}}^{\frac{N}{2}-1} f[n_1, n_2] e^{-\frac{2\pi i n_1}{N} a k_1} e^{-\frac{2\pi i n_2}{N} b k_2}$$

Where a, b are predefined scalars. Since GIFT has two adjustable parameters it's more flexible.
$F^{a,b}[u, v] = F[au, bv]$ ([20],p.2)

GIFT grid was defined in similar way like MRFLT grid. For GIFT Cut is defined not a sequence of integers, bus as a sequence of (a_i, b_i) tuples.

3.2.4.4 Radon Transform with MRFLT for line detection

Paper [1] doesn't contain details of implementation, but it referenced to [20],[14]. Authors of [20] in their work also give credits [16]. By their experiments GIFT line detector can compete with RANSAC.

The algorithm is the following:

1) Input image – is grayscale image
2) Set the adjustable parameter L and Cut of GIFT. Authors suggest to concentrate in high frequencies during detecting edges. $Cut = \{(a_i, b_i): 0.5 < a, b < 1, i = \overline{1, L}\}$
3) Compute series of grids of Fourier spectrum for each layer, combine all layers. As output we have L-layer GIFT
4) Discretize $\theta = \{0, d\theta, 2d\theta, \ldots, \pi\}$
5) Discretize $v = \{0, dv, 2dv, \ldots, \frac{\sqrt{r^2+s^2}}{2}\}$ as much as possible, where r,s – size of 2d Fourier spectrum
6) Use nearest or bilinear interpolation to compute the frequency at a position $x = vcos(\theta), y = x = vsin(\theta)$
7) Right now we're ready to compute IDFT along radial directions. But before it extra enhancements can be done like smoothing filters.
8) Apply 1D IDFT per each fixed angle θ
9) Find peaks in sinogram. When peak is identified remove it from sinogram with surround peaks. It is also comfortable to know number of straightlines to search.

Experiments from [20] show that L=3 lead to well results.

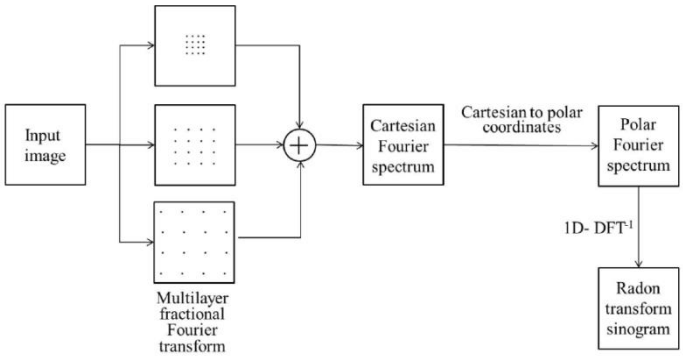

Fig.2. Schematic description of generating sinogram for find peaks.

3.2.5 Post-process. Sampling detected lines

From previous steps we have bird's eye view of the scene and candidate lane lines. (Fig 3.a). Each of this line is sampling via green columns (Fig 3.b).

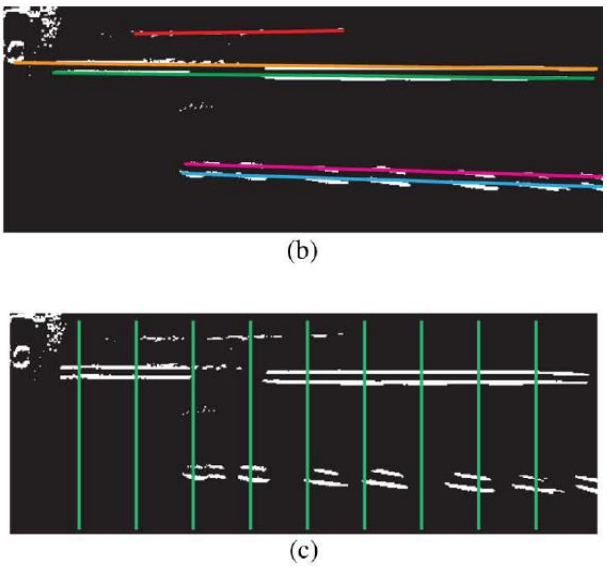

Figure 3. Lane candidates in birds eye view space

3.2.6 Post-process. 1D template matching

Near each sampled point from previous step we setup a window. In this window we do template based detection. Slide a subwindow in and compute similarity with template. Similarity in this approach measured by Normalized cross correlation(NCC). This NCC score between two signals w and w' is evaluating as follows $\left(\frac{(w-\bar{w})^T(w'(u)-\overline{w'})}{|w-\bar{w}|\,|(w'(u)-\overline{w'})|}\right)$

I think that template matching have been chosen because in principle it is good recognition method, when there are no so big intra-class variation. And it's our situation.

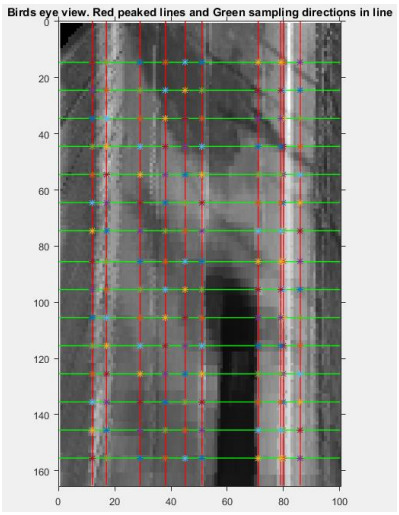

Figure 4. Example of sampled points for template matching

Authors of [1] didn't explain how they performed filtering. I assumed that 15% of finding points correspond to false peaks.

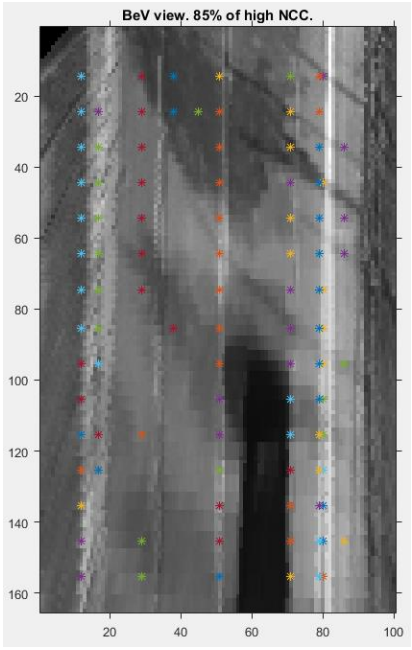

Figure 5. Example of result after 1D template matching

3.2.7 Post-process. RANSAC and LSE

Output from 1.8 is a set of lane markers candidates. Use RANSAC – RANdom Sample Consensus (Fisher, Boles, 1981) to eliminate outliers:

1) Separate observed points into two sets/find participation P which participate set of points πinto two sets P->{I, O}, O – outlier set, I – inlier set
2) We want minimize size of the outlier set
3) We push extra constraint that residual error is at least less then threshold delta $r(I,h) < \delta$ where h is parameter of our model, and δ us fixed at the beginning, parameter of the fitting

After rejecting the outliers, linear least squares estimation (LSE) is used to find the best fit of a line to the inliers.

For $Ax = b$ and the discrepancy of the system $d = b - Ax$ for problem of $d \cdot d^T$ minimization solution exist in closed-form.

To minimize this we should chose $x = (A^T A)^{-1} A^T b$. This equation is called "normal equation" and can be derived:

1) Via Linear Algebra looking into $H = span\{a_1, \dots, a_k\}$ and $H^\perp$ which is orthogonal complementary of H to $\mathbb{R}^n$. Then look into $b = h + h^\perp, h^\perp \epsilon H^\perp, h \epsilon H$. Minimum discrepancy was reached by x was defined via $Ax = h$.
2) Or via minimization of Euclidian norm of $Ax - b$ $\nabla\left(||Ax - b||^2\right) = 0$

In our context:
B – distance to the line

X – parameters of 2d line
A – points in 2d bird eye view space written by rows

4. Experiments

During development it was observed that MLFFT method has more accurate peaks then during using simple Radon transform. I implemented all algorithms for this work in Matlab. I also want to mention that real bottleneck stage is MLFFT step is very time consuming due a lot of different mathematics in it.

In my experiments when blur introduce noise rather huge MLFFT step absolutely ignore real peaks. So blurring should be take with care.

Also after evaluation experiments even with straight lanes is was observed that this tuned threshold parameters approximately for all steps can significantly affect into result image:

- Number of images to blur
- Size of Birds Eye View image
- MLFFT grid parameters
- Number of peak lines after Radon Transform with MLFFT
- Template matching threshold of dropping after 1D template matching
- Ransac parameter for remove outliers
- Number of lines in the road.

Now it was realized that even approach from work [1] is very interesting, right now in each step there are a lot of tune parameters are presented.

During using algorithm sometimes result images looks like in original paper. Sometimes more noisy.

Figure 6.Result images

Current implementation is in matlab and is not real time and assume some input data such as ROI and other configurable options. During my experiments average error in estimating lanes to the real location in ROI is ~30pixels, where the size of frames from KITTI dataset are 1242x375.
Test images and camera parameters are available here: *http://www.cvlibs.net/datasets/kitti/eval_road.php*

For using camera intrinsic and extrinsic parameters this expression can be used:
$x = P_2 R_{0,rect} TrCamToRoad^{-1} X$, whew
P_2 - projection matrix for left color camera 3x4
$R_{0,rect}$ $TrCamToRoad$ – matrices that should be extended by 4x4
X – point world coordinates
x – image space homogeneous coordinates.

Unfortunately, author in paper [1] mentioned only speed of their implementation, but didn't say anything about accuracy of their original solution.

5. Conclusion

In theory approach presented in [1] is perfect. But during performing experiments it was observed that in implementation there a lot of parameters need to be tuned, so in the future for make algorithm adaptive extra work need to be done.

I hope to port this work into C++ , use it in my future, and make it realtime.

I think humans can compete with Neural Networks only via finding interesting things in functional analysis, where the dim of the space if not finite at all. It opens the doors to concepts "understandable" for human but really not for computer.

During reading references I have seen that Fourier based method can also be used for finding Shift, Rotate, Scaling invariant features.

6. References

Authors, Title, Year
[1] Payam S. Rahmdel, Daming Shi, and Richard Comley: "Lane detection using Fourier-based line detector", 2013
[2] Mohamed Aly: "Real time Detection of Lane Markers in Urban Streets", 2008
[3] Heechul Jung,Junggon Min, Junmo Kim: "An Efficient Lane Detection Algorithm For Lane Departure Detection", 2013
[4] Jie Guo, Zhihua Wei, Duoqian Miao: "Lane Detection Method Based on Improved RANSAC Algorithm",2015
[5] Yong Chen, Mingyi He: "Sharp Curve Lane Boundaries Projective Model and Detection", 2012
[6] Jianyu Yang, Zhuo Li, Liangchao Li: "Lane Detection Based on Classification of Lane Geometrical Model",2012
[7] Huachun Tan, Yang Zhou, Yong Zhu, Danya Yao, Keqiang Li: "A novel curve lane detection based on Improved River

Flow and RANSAC",2014
[8] Abdulhakam.AM.Assidiq, Othman O. Khalifa, Md. Rafiqul Islam, Sheroz Khan, "Real Time Lane Detection for Autonomous Vehicles", 2008
[9] Amol Borkar, Monson Hayes, Mark T. Smith, Sharathchandra Pankanti: "A Layered Approach To Robust Lane Detection At Night",2009
[10] Bracewell R.: "Fourier Analysis and Imaging",2003
[11] Bracewell R.: "The Fourier Transform and Its Applications 3rd ed.",2000
[12] Sebastian Thrun, Wolfram Burgard, Dieter Fox: "Probabilistic Robotics.",2005
[13] Amol Borkar, Monson Hayes, Fellow, IEEE, and Mark T. Smith: "A Novel Lane Detection System With Efficient Ground Truth Generation",2012
[14] D. Shi, L. Zheng, and J. Liu: "Advanced Hough transform using a multilayer fractional Fourier method",2010
[15] S. R. Deans, "Hough transform from the Radon transform",1981
[16] Wei Pan, Kaihuai Qin, and Yao Chen: "An Adaptable-Multilayer Fractional Fourier Transform Approach for Image Registration",2009
[17] B. Srinivasa Reddy and B. N. Chatterji: "An FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration",1996
[18] William L. Briggs, Van Emden Henson: "The DFT: An Owners' Manual for the Discrete Fourier Transform",1995
[19] Edited by Ales Ude: "Robot Vision",2010
[20] L. Zheng and D. Shi, "Advanced Radon transform using generalized interpolated Fourier method for straight line detection",2011

7. Appendices

7.1 Radon Transform

7.1.1. From point impulse to line impulses.

Let's take Dirac's delta function which is function of one scalar variable. Then evaluate dot product nx in 2 dimensional Cartesian space.
$\delta_n(x) = \delta(nx)$ – it's a definition of impulse along a line n which go through origin. Right hand side is usual delta-function. $\delta_n(x - cn)$ – impulse along a line which is not necessary go though origin .
$\delta(nx - c)$ – alternative notation for impulse along a line.

There are two approaches to understand what is going on. First is that $\delta(nx - c)$ is the sequence of functions like $\frac{1}{\varepsilon}\Pi_{\varepsilon n}(x - cn)$. Via this approximation we can derive that $\int_{\mathbb{R}^2} \delta(nx - c)f$ becomes an integral of $\boldsymbol{f}$ over the curve in a function domain which is line. This equation you should understand in average sense, not pointwise. Another approach is to look at $\delta(nx - c)$ from point of view distribution. Distribution is complex valued, linear, continuous operators on test functions, it's not a distribution from probability theory.

7.1.2 Tomography and Radon Transform

Tomography (Tomas – by Greek means section) - reconstruct 2d image from knowledge of 1D sections. Pass a light from a tube. We know I_{in} and I_{out}. For part of the tube with length dx we assume that light change intensity can be described via:
$\frac{dI}{I} = -\mu(x)dx$, where μ is murkeiness coefficient, which we want to understand.

$$\Rightarrow \quad I_{out} = I_{in}e^{-\int_{tube}\mu(x)dx}$$

$$\Rightarrow \quad -\ln(\frac{I_{out}}{I_{in}}) = \int_{tube}\mu(x)dx$$

Now Let's take 2D slice of something, e.g. your body. Density can be described by scalar function $\mu(x_1, x_2)$. We want to know it. Maybe the integrals along lines are enough to determine $\mu(x_1, x_2)$ everywhere. Remarkably the answer is yes, but it's not obvious!

Steps to solution:

1) Formulate a problem as transform problem. Fix $\mu(x_1, x_2)$ and know with a line "L" we can associate a number $\int_{line}\mu(x_1, x_2)$. Is was called **Radon Transform.** $\boldsymbol{R}_\mu(L) = \int_L \mu$. It is a scalar function of line. It was introduced by Johann Radon in 1917 year. It was assumed that $\mu(x_1, x_2)$ is equal to zero outside it's domain.

2) Encode all lines in the plane.
$$\rho - xcos(\theta) - ysin(\theta) = 0,$$
$$0 \le \theta < \pi, -\infty \le \rho \le +\infty$$

3) Use concept of line impulse from previous section:

$$\delta_L(x, y) = \delta(\rho - xcos(\theta) - ysin(\theta))$$

Via this thing following integral produce integral along a line, not pointwise, but in average sense:
$\iint_{\mathbb{R}^2} f\delta_L dx_1 dx_2 = \int_L f$ =>

$$\boldsymbol{R}_\mu(L) = \int_L \mu = \iint_{\mathbb{R}^2} \mu(x_1, x_2)\delta_L dx_1 dx_2$$

$$= \iint_{\mathbb{R}^2} \mu(x_1, x_2)\delta(\rho - x_1 cos(\theta) - x_2 sin(\theta))dx_1 dx_2$$

This expression

$$R_\mu(\theta,\rho) = \iint_{\mathbb{R}^2} \mu(x_1,x_2)\delta(\rho - x_1 cos(\theta) - x_2 sin(\theta))dx_1 dx_2$$

is called **Radon Transform in coordinates**. It can be proved that this transform is linear, and result of Radon transform is always even function ($R_\mu(\theta,\rho) = R_\mu(\pi+\theta,-\rho)$)

4) If look at lines with fixing θ and allow ρ to vary then we will look at family of parallel lines with normal direction which make angle θ with OX axe. Collection of values $R_\mu(\theta,\rho)$ for this series of parallel lines referred to as **Projection of μ**

7.1.3 Examples of Radon Transform

$$circ(r) = \begin{cases} 1, r \le 1 \\ 0, r > 1 \end{cases}$$

$$(Rcirc)(\theta,\rho) = \begin{cases} 2\sqrt{1-\rho^2}, |\rho| \le 1 \\ 0, |\rho| > 1 \end{cases}$$

$$R\left(e^{-\pi({x_1}^2+{x_2}^2)}\right) = e^{-\pi\rho^2}$$

7.1.4. Radon Transform interconnection with Fourier Transform

$$\boldsymbol{R}_\mu(L) = \boldsymbol{R}_\mu(\theta,\rho) = \iint_{\mathbb{R}^2} \mu(x_1,x_2)\delta(\rho - x_1 cos(\theta) - x_2 sin(\theta))dx_1 dx_2$$

Let's fix θ, and take 1D Fourier Transform w.r.t. to ρ. Physically it means look at family of parallel beams through the region.

$$\mathcal{F}\boldsymbol{R}_\mu(r,\theta) = \int_{-\infty}^{+\infty} e^{-2\pi i r\rho}\left(\iint_{\mathbb{R}^2} \mu(x_1,x_2)\delta(\rho - x_1 cos(\theta) - x_2 sin(\theta))dx_1 dx_2\right)d\rho$$

$$= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \mu(x_1,x_2)\,dx_1 dx_2 \int_{-\infty}^{+\infty} e^{-2\pi i r\rho}\,\delta(\rho - (x_1 cos(\theta) + x_2 sin(\theta)))d\rho$$

$$\mathcal{F}\boldsymbol{R}_\mu(r,\theta) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \mu(x_1,x_2)\,(e^{-2\pi i r(x_1 cos(\theta)+x_2 sin(\theta))})dx_1 dx_2$$

Use r,θ as polar coordinates in ζ_1,ζ_2 cartesian plane coordinate system.

$\zeta_1 = rcos(\theta) \quad \zeta_2 = rsin(\theta)$

${\zeta_1}^2 + {\zeta_2}^2 = r^2 \quad tan(\theta) = \frac{\zeta_2}{\zeta_1}$

=>

$$(\mathcal{F}(R_\mu, and\ fix\ \theta))(r,\theta) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \mu(x_1,x_2)\,dx_1 dx_2\left(e^{-2\pi i(x_1\zeta_1 + x_2\zeta_2)}\right) = \mathcal{F}_{2d}\mu(\zeta_1,\zeta_2)$$

This means that via 1d Fourier transform w.r.t ρ and fixed θ we know 2d Fourier Transform of μ in ζ_1,ζ_2 points. And we have equation that point $(\zeta_1,\zeta_2) = (rcos(\theta), rsin(\theta))$. Even we fixed θ, we can iteratively change θ, find new set of points $(\zeta_1,\zeta_2) = (rcos(\theta), rsin(\theta))$ and evaluate Fourier Transform in it.

Let's call series of 1D Fourier transform as G: $G(r,\theta) = (\mathcal{F}(R_\mu, and\ fix\ \theta))(r,\theta)$. We *approximately* know $\mathcal{F}_{2d}\mu(\zeta_1,\zeta_2)$. The last step is to invert it to receive $\mu(x_1,x_2)$

Such process of reconstructing μ from $\mathcal{F}_{2d}\mu$ is called **Projection Slice Theorem or Central Slice Theorem.** "*approximately"* because a problem is in that: G is defined in polar coordinates, but inverse Fourier Transform should be done in Cartesian grid.

Technic for creating regular grid from polar grid is **gridding**.

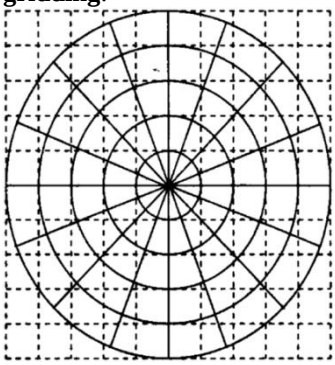

Fig 7. Polar and Cartesian grid

7.1.5.Properites of Radon Transform

$$f(\theta,p) = R_\mu(L) = \int_L F = \iint_{\mathbb{R}^2} F(x_1,x_2)\delta_L dx_1 dx_2 = \iint_{\mathbb{R}^2} F(x_1,x_2)\delta(\rho - x_1 cos(\theta) - x_2 sin(\theta))dx_1 dx_2$$

Properties which was mentioned in [15]

1. If $F = \delta(x-x_0)\delta(y-y_0)$ => f is non-zero in sinusoidal curve $p = x_0 cos\theta + y_0 sin\,\theta$
2. A point (θ_0,p_0) in the (θ,p) plane corresponds to line $\rho_0 = x_1 cos(\theta_0) + x_2 sin(\theta_0)$
3. Each of collinear points in x_1,x_2 mapped to sinusoidal curve, but they are intersect in the (θ_0,p_0)
4. Points in curve $p = x_0 cos\theta + y_0 sin\,\theta$ in the (θ,p) plane correspond to lines which all pass through (x_0,y_0)

The 2D Radon transform is usually graphically represented as a sinogram, which means the intensity value in (θ,p) plane. Also Radon Transform has interesting properties: linearity, special shifting, rotation, scaling theorem, convolution theorem.

7.2 Fourier Transform and DFT

7.2.1. 1D DFT, 2D DFT Definitions

Forward FT: $\mathcal{F}f(s) = \int_{-\infty}^{+\infty} e^{-2\pi i s t} f(t)dt$

Inverse FT: $(\mathcal{F}^{-1}g)(t) = \int_{-\infty}^{+\infty} e^{2\pi i s t} g(s)ds$

Forward 1d DFT: $Ff[n] = \sum_{k=0}^{N-1} f[k]e^{-\frac{2\pi i n k}{N}}$

Inverse 1d DFT: $F^{-1}f[n] = \frac{1}{N}\sum_{k=0}^{N-1} f[k]e^{\frac{2\pi ink}{N}}$

Forward and inverse 2d FT:

$$\mathcal{F}f(\zeta_1,\zeta_2) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x)e^{-2\pi i(x_1\cdot\zeta_1+x_2\cdot\zeta_2)}dx_1dx_2$$

$$\mathcal{F}^{-1}f(x_1,x_2) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x)e^{2\pi i(x_1\cdot\zeta_1+x_2\cdot\zeta_2)}d\zeta_1d\zeta_2$$

Aout high dimensional frequencies:
$e^{\pm 2\pi i(x\cdot\zeta)} = 1$ iff $x\cdot\zeta$ is an integer. Let's fix ζ, and assume we worked in $\mathbb{R}^2$ space =>
If $x_1\cdot\zeta_1 + x_2\cdot\zeta_2 = i$, where i is an integer =>
$e^{\pm 2\pi i(x\cdot\zeta)} = 1$, i.e. it is equally spaced lines with

distance between each other $\frac{1}{\sqrt{{\zeta_1}^2+{\zeta_2}^2}}$, and with normal (ζ_1,ζ_2). When people say high frequency it means high magnitude of $||\zeta||$. Similar idea is in more high dimensions, more then 2.

Forward and Inverse 2d DFT ([18], p.151,p.23):

$$Ff[j,k] = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f[m,n]e^{-\frac{2\pi imj}{M}}e^{-\frac{2\pi ink}{N}}$$
$$F^{-1}f[m,n] = \frac{1}{MN}\sum_{j=0}^{M-1}\sum_{k=0}^{N-1} f[j,k]e^{\frac{2\pi imj}{M}}e^{\frac{2\pi ink}{N}}$$

[18]p.162. It is possible to view at 2d DFT as two-phase evaluation. Computational cost of FFT is NM(logM + logN)

7.2.2. 1D DFT insights

Assume: $f(t)$ is zero outsize [0,L]
$\mathcal{F}f(t)$ is zero outsize [0,2B]
But:

1) Such constraints are **lie.** A bit tricky, but it is possible to prove that non-zero signal can not be both time-limited and band-limited.
2) L and B are integers – **is one more strange thing**, but allowable
3) One more strange thing as you will see **there is no any negative frequencies.** But it's a problem of shifting.

After applying sampling Theorem (slight modification of it due to non symmetrical interval) we can get:
$N = 2BL \quad t_n = \frac{n}{2B}, n = \overline{0,N-1} \;\; s_k = \frac{k}{L}, k = \overline{0,N-1}$

$$\mathcal{F}f_{discr}(s_k) = \sum_{n=0}^{N-1} f(t_n)e^{-2\pi is_kt_n} = \cdots \approx \frac{1}{2B}\mathcal{F}f(s_k)$$

So in some sense $\mathcal{F}f_{discr}$ after different wrong statements now is an approximation of $\mathcal{F}f(s_k)$ but up to factor $\frac{1}{2B}$
Moreover $e^{-2\pi is_kt_n}$ can be transformed into $e^{-2\pi is_kt_n} = e^{-2\pi i\frac{k}{L}\frac{n}{2B}} = e^{-\frac{2\pi ikn}{N}}$

DFT can be rewritten into a form $w[n] = e^{\frac{2\pi in}{N}}$ => $\overline{w[n]} = e^{-\frac{2\pi in}{N}}$ =>
DFT(forward): $Ff[n] = \sum_{k=0}^{N-1} f[k]e^{-\frac{2\pi ink}{N}} = \sum_{k=0}^{N-1} f[k]w[n]^{-k}$

7.2.3. Two grids of DFT

In the time domain the signal is limited to an interval of length L. In the frequency domain the transform is limited to an interval of length 2B. The grid points in the time domain are spaced 1/2B apart. The grid points in the frequency domain are spaced 1/L apart. So the spacing in one domain is determined by properties of the function in the other domain

$$\Delta t = \frac{1}{2B}, \Delta\nu = \frac{1}{L}, \Delta t\Delta\nu = \frac{1}{N}$$

7.2.4. DFT. Where is negative frequencies?

Formulas represented here areequal to formulas which are sed in matlab to evaluated DFT via FFT.

Let's assume for example N=8. The result tuple of the DFT is: $[Ff(0), A, B, C, D, \bar{C}, \bar{B}, \bar{A}]$
Value at index N/2 if count indicies from zero split interval frequencies into positive and negative.
It can be shown that value at N/2 position for real value samples are real. Value at index 0 is DC component.
Matlab contains command *fftshift* to convert mentioned tuple into $[\bar{C}, \bar{B}, \bar{A}, Ff(0), A, B, C, D]$ So if not apply *fftshift* then after DFT the situation was the following: [DC,L, H,…,H,L]. Similar situation is with 2D DFT.

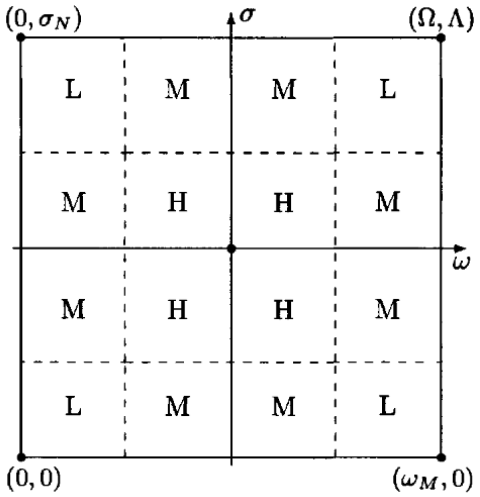

Fig.8 Picture from [18],p. 152