



UNIVERSIDAD DE MONTERREY

Reporte de actividades

Proyecto de Evaluación Final

Pedro Aguiar

2/Mayo/2014

Índice

1. Minuta 01	3
1.1. Proceso Administrativo	3
1.2. Actividades	3
1.2.1. PCB comunicación entre cámara y raspberry pi	3
1.2.2. Alternativa a Microchip	4
1.2.3. Estudio e implementación en C de transformada de Hough	5
1.2.4. Revisión bibliográfica de aplicaciones de la transformada de Hough.	6
1.3. Resultados	6
1.3.1. PCB comunicación entre cámara y raspberry pi	6
1.4. Alternativa a Microchip	6
1.5. Estudio e implementación en C de transformada de Hough	7
1.5.1. Revisión bibliográfica de aplicaciones de la transformada de Hough.	7
1.6. Pendientes	7
2. Minuta 02	8
2.1. Actividades	8
2.2. Resultados	8
2.3. Pendientes	8

Índice de figuras

Índice de tablas

1.1. Bill of materials para el PCB de conexión entre el PIC y la cámara. . . .	4
--	---

1 Minuta 01

23 de Abril - 2 de Mayo

1.1. Proceso Administrativo

En esta semana todavía hubo pendientes administrativos que se reportan a continuación, como referencia e información para futuros estudiantes de la UDEM que vayan a participar en un programa similar. Al llegar al laboratorio lo primero que hicieron fue llevarme con Jennyfer DUBERVILLE para revisar los procedimientos administrativos. Se revisó que la papelería que había sido enviada por internet estuviera correcta. La papelería consistió en:

- Credencial de estudiante
- Copia del último certificado (en mi caso el de la preparatoria)
- Copia del pasaporte
- Un contrato (firmado por mí y por el director de DIT, por parte de la UDEM)

Después de corroborar la papelería se le entregó un formulario a mi asesor y a mí me dieron un reglamento acerca del uso de los recursos del laboratorio (internet, equipo electrónico, etcétera). Se me hizo firmar un papel en el que decía la fecha de llegada al laboratorio y donde declaraba haber leído el reglamento. Se me recordó que los primeros días de Mayo tendría que pagar las cuotas de "social securityz civil liability". También se me informó que para recibir la gratificación tengo que abrir una cuenta en un banco Francés y me dijeron que me iban a contactar con una persona que podía ayudarme con eso.

Por último se me proporcionaron las credenciales de acceso a Internet y mi computadora. Se instaló el equipo de cómputo que se me asignó y se me explicó cómo utilizarlo, así como algunos comentarios finales (no se puede conectar nada a la red alámbrica, no puedo instalar software que no esté en los repositorios de Debian, cómo acceder a mi espacio en el servidor de la escuela que está respaldado, detalles de ese tipo).

1.2. Actividades

1.2.1. PCB comunicación entre cámara y raspberry pi

Durante el primer día de trabajo se hizo un bill of materials (Tabla 1.1) para hacer una tarjeta de circuito impreso que conecte una cámara OV7670 con el raspberry, el objetivo siendo que el Carrito tenga capacidades de visión computarizada. La diferencia con la de la UDEM es que Varrier externó que le gustaría recibir en el raspberry:

- Máxima resolución (640x480).
- Máximo framerate (30fps).

Componente	Cant.	Precio Unitario	Costo (euros)
PIC18F45K80-I/P	1	3.46	3.46
Capacitor cerámico 0.1 uF	10	0.386	3.86
Capacitor cerámico 1 nF	25	0.114	2.85
Capacitor tantalum 10 uF	5	0.822	4.11
Capacitor cerámico 15 pF	10	0.262	2.62
Resistor 470 ohm	10	0.034	0.34
Resistor 10k	10	0.025	0.25
Resistor 330	10	0.034	0.34
Resistor 470	10	0.034	0.34
Cable (jumper terminals)	30	0	0
Headers (pins)	2	3.43	6.86
Quartz	1	2.11	2.11
Switch SPDT	3	2.27	6.81
PicKit2	1	33.99	33.99
		Subtotal	67.94
		Total	81.528

Tabla 1.1: Bill of materials para el PCB de conexión entre el PIC y la cámara.

- Imagen a color.

Observaciones sobre el bill of materials:

- Costo elevado
 - 40.74 euros de componentes de una placa.
 - 33.99 euros para el programador.
- El proveedor impone cantidades mínimas de componentes.
- No está considerado el costo de mandar a hacer la placa.
- No se podría utilizar el microcontrolador de la UDEM porque no estaban considerados los bits para el color.
- Por lo tanto, el PCB tendría que ser diseñado completamente desde cero con la sensación (personal) de menos herramientas y proveedores menos accesibles.

1.2.2. Alternativa a Microchip

Debido a lo costoso en dinero (componentes) y tiempo (diseño y armado) que resultó ser el diseño de un PCB para establecer la comunicación con las cámaras procedí a buscar una alternativa que ofrecerle a Varrier. La mejor alternativa fue utilizar un STM32, porque:

- Tienen Digital Camera Module Interface (DCMI).

- Tienen Direct Memory Access (DMA).
- Combinando DCMI+DMA se pueden guardar frames en memoria usando sólo hardware (el procesador queda libre para usarlo para otras aplicaciones).
- Hay tarjetas de evaluación de bajo costo (una de 12 y otra de 21 euros).
- Programación por USB.
- Varrier tiene experiencia y confianza en estos dispositivos.

Se investigó para validar que las tarjetas de evaluación pudieran hacer la comunicación con la cámara. También se revisó si se podrían utilizar los sensores y la pantalla LCD de la tarjeta (si no interferían con los pines que se necesitaban para la cámara). Se tomó la decisión de comprar dos tarjetas 32F429IDISCOVERY.

1.2.3. Estudio e implementación en C de transformada de Hough

El código implementado en la UDEM para detectar la línea que se está siguiendo utiliza OpenCV, una librería que tiene una gran variedad de algoritmos para hacer visión computarizada. Es un objetivo del proyecto que el Carrita pueda medir la distancia que hay hacia un vehículo que esté enfrente y esta función no estaba implementada.

Antes de comenzar la implementación decidí deshacerme de la dependencia de OpenCV y desarrollar código en C para una transformada de Hough basado en los siguientes hechos:

- OpenCV es una librería muy generalizada con una gran cantidad de funciones (la descarga de la versión estable actual, 2.4.9 es de más de 87MB para Linux).
- Para la detección de línea sólo se estaban usando dos "funciones complejas" de OpenCV.
 - Canny edge detector.
 - Hough transform.
 - Seguramente menos del 0.5% de los 87MB.
- La transformada de Hough de OpenCV no puede usarse para detectar al vehículo de enfrente (sólo se detectan líneas rectas y círculos).
- OpenCV hace el cálculo completo de la transformada en cada frame. Nuestra aplicación puede aprovechar el hecho de saber que es un video para no realizar el cálculo completo en cada frame y buscar alrededor del punto conocido del cálculo anterior.

Además tenemos las ventajas típicas de programar una versión independiente de librerías:

- Conocimiento profundo de la transformada (funcionalidad menos abstracta).

- Adaptación específica a la necesidad (minimalista, rápida, a la medida).
- Libertad en funcionalidad, optimizaciones e interfaces.

En algún momento se iba a llegar al límite de la capacidad del raspberry utilizando OpenCV y para empujar ese límite se tendría que bajar a una solución específica sin uso de librerías generalizadas.

Para implementar en C la transformada de Hough se llevaron a cabo las siguientes tareas:

- Estudio de la teoría detrás de la transformada de Hough.
- Búsqueda y comparación de código existente para la transformada.
- Adaptar una versión a usar sólo C (sustituir librerías).
- Implementar optimizaciones
 - (Ab)uso del preprocesador (para reducir bifurcaciones).
 - Aritmética de punto fijo.
 - Tablas de funciones trigonométricas.
 - Memoria asignada estáticamente.
 - Region de interés.
 - Reducción de grados de libertad.

Se utilizó SDL para dibujar en pantalla (no es necesario, pero sirve para ver y comprobar resultados) y se adaptó fast-edge (<http://goo.gl/ARWJZj>), un proyecto encontrado en internet, para hacer detección de bordes.

1.2.4. Revisión bibliográfica de aplicaciones de la transformada de Hough.

Se han descargado 25 artículos relevantes al tema en los que se están observando aplicaciones de extracción de características usando la transformada de Hough. Al seleccionarlos se observó que hablan de las limitantes de la transformada, pequeñas modificaciones para resolverlas, así como propuestas y resultados experimentales.

1.3. Resultados

1.3.1. PCB comunicación entre cámara y raspberry pi

- Bill of materials de la tabla 1.1.
- Debido al costo elevado en tiempo y dinero se decidió buscar una alternativa.

1.4. Alternativa a Microchip

- Se utilizará un STM32.
- La tarjeta a utilizar es la 32F429IDISCOVERY.

1.5. Estudio e implementación en C de transformada de Hough

- Se entendió toda la teoría detrás de la transformada.
- Se implementó la funcionalidad de la transformada de Hough tradicional utilizando sólo C.

1.5.1. Revisión bibliográfica de aplicaciones de la transformada de Hough.

- Empecé una revisión bibliográfica
- Observé las modificaciones (y su justificación) que se hacen a la transformada de Hough "tradicional" para implementarla en aplicaciones reales.

1.6. Pendientes

- Gantt: Le puedo adelantar que Varrier y yo estuvimos de acuerdo en que en dos meses tenemos que haber hecho el control de distancia (y de ser posible también lateral) utilizando las cámaras y haber enviado un artículo a alguna conferencia. Después de esos dos meses me dijo que lo decidiríamos después en base a mis intereses, etcétera. La única opción que le ofrecí fue la de simular la flotilla de vehículos y le pareció buena idea pero me dijo que lo siguiera pensando y cuando estuviera la fecha más cercana tomábamos la decisión. Entiendo algo como que el plan de colaborar con otro equipo (que se iba a encargar de la simulación de la flotilla) se vino abajo.
- Compras: Varrier me estimó que la próxima semana estarán llegando los componentes.
- Repositorio y Dropbox: Estuve buscando una forma de sincronizar y respaldar el trabajo, ya me rendí y parece que no habrá formas cómodas (por restricciones de no instalar software que no esté en los repositorios de Debian y debido a que estamos atrás de un proxy que necesita contraseñas), pero aun así subiré todo el trabajo a algún lugar, le aviso la decisión en el próximo reporte.
- Las actividades se dibujan así: programación de STM32, trabajo electrónico para enviar las imágenes de la cámara al raspberry, desarrollo de transformada de hough específica para la aplicación, sacar modelos adecuados para la aplicación, implementar control tolerante a fallas.

2 Minuta 02

5 de Mayo - 9 de Mayo

2.1. Actividades

2.2. Resultados

2.3. Pendientes