

Corrector automático de plantillas test con teléfono móvil, tableta o cámara web

Iván Pascual Palacios

Resumen—Este proyecto consiste en el desarrollo de una corrector automático de plantillas test mediante la realización de una fotografía. En particular para este proyecto se ha diseñado una plantilla específica que tendrá que usar el profesor para poder corregir los exámenes automáticamente. Se ha restringido el uso a una única plantilla muy específica para intentar ofrecer unos resultados lo mas fiables posibles para el usuario que utilice dicha aplicación. Para garantizar estos buenos resultados tendrá que seguir una serie de pasos explicados en este documento.

Palabras clave—detección de corners, Algoritmo de Harris, Matlab, transformada geométrica, correspondencia

Abstract—This project consist in the development of an automatic correction of test templates by a photography. Particularly for this project has been designed a specific template that will have to use the teacher to do the automatic correction of the exams. This application has been restricted to a single very specific template to try to provide a more realiable results for the user of the application. To ensure this succes will have to follow certain steps outlined in this document.

Index Terms—corner detection, Harris algorithm, geometric transform, matching

1 INTRODUCCIÓN

En muchas ocasiones la labor de corregir exámenes tipo test, acostumbra a ser muy monótona y repetitiva para los profesores. El objetivo principal es crear una aplicación que pueda corregir un examen fotografiado desde un teléfono móvil utilizando unas plantillas de exámenes tipo test determinadas. Dicha aplicación presenta una gran serie de dificultades que la hacen realmente compleja para asegurar una fiabilidad alta.

Las aplicaciones del campo de la visión por computador y procesamiento de imágenes tienen la dificultad añadida de que se basan en la gran casuística del mundo físico, en este caso dicha dificultad radicará en factores como la calidad de la imagen, la iluminación, la presencia de ruido, la inclinación, la perspectiva, el enfoque de la fotografía, etc.

En este proyecto la aplicación se ha realizado para escritorio, pese a que la idea inicial era realizarla para dispositivos móviles se optó por la utilización de un lenguaje de alto nivel para resolver el problema y en un futuro poder adaptar dichos algoritmos, en una versión optimizada para dispositivos móviles. El documento contiene los siguientes apartados:

Estado del Arte, metodología, detección de corners, transformada geométrica, resultados, trabajos futuros y conclusiones.

2 ESTADO DEL ARTE

Actualmente existen correctores de exámenes tipo test utilizados en diversos sectores, no sólo en la educación, pero en la mayoría de los casos se basan en plantillas de exámenes muy limitadas y poco intuitivas.

Gran parte de los proyectos de este ámbito actualmente utilizan una maquinaria especialmente diseñada para corregir estos exámenes, utilizando técnicas de escaneo, pero no de procesamiento de imágenes y de visión por computador ya que éstas presentan una complejidad mucho más elevada y una menor fiabilidad.

Los proyectos que utilizan visión por computador también utilizan plantillas muy limitadas y presentan unos resultados poco fiables cosa que hace que muchas empresas y sectores educativos opten por comprar maquinaria especialmente diseñada para corregir plantillas de exámenes tipo test. Aquí está la referencia al corrector de exámenes tipo test:
http://iris.sel.eesc.usp.br/wvc/Anais_WVC2013/Poster/1/9.pdf

3 METODOLOGÍA

Este proyecto lo he llevado a cabo siguiendo la metodología espiral que es un tipo de metodología ágil muy utilizada actualmente.

Para poder resolver los problemas surgidos y llevar a cabo este proyecto he seguido las siguientes fases:

1. En esta primera fase se plantea el problema a resolver, se discute con el profesor cuales son las mejores maneras de afrontar el problema e intentar buscar su solución. También es una fase de

- E-mail de contacto: pascualpalaciosivan@gmail.com
- Mención realizada: Computació
- Trabajo tutorizado por: Javier Sánchez Pujadas (CVC)
- Curso 2014/15

- investigación de los diferentes algoritmos básicos que se utilizan en el proyecto.
2. La dificultad de este proyecto y en muchos otros de visión por computador radica en encontrar un método genérico para poder resolver el mismo problema con imágenes muy diferentes en cuanto a resolución, iluminación, marcas del alumno etc. Por tanto en esta fase nos centramos en la parte fundamental del proyecto que es la de preprocesar la imagen , es decir, binarizarla y filtrar por área para eliminar el ruido de la imagen (módulo procesamiento de imágenes).
 3. Una vez se han realizado pruebas con imágenes de tipos muy diferentes y se ha comprobado que el preprocesado es correcto y funciona con un gran porcentaje de imágenes pasamos a la fase de tratamiento de la imagen, es decir, buscar los corners de las dos tablas que nos servirán como muy buenas referencias para poder superponer las dos imágenes (módulo de búsqueda de corners).
 4. Aplicar la transformación geométrica entre la imagen del alumno y la original para que quede lo más bien orientada posible (módulo de transformada geométrica).
 5. Encontrar los centros de las celdas de las dos tablas y verificar si existe alguna marca del alumno en dichas celdas y si es así clasificarlas según su posición (módulo extracción respuestas del alumno).
 6. Aplicar las opciones previamente entradas por el usuario donde especifica las respuestas correspondientes del examen y se procederá a validarlas
 7. Testear cada una de las fases por separado, es decir, fase de preprocesado, fase de búsqueda de corners, transformación geométrica y búsqueda de las respuestas en las tablas.
 8. Realizar la primera versión de la interficie.
 9. Testear la interficie con las diferentes opciones y anotar los resultados con todas las imágenes de prueba.

4 PLANIFICACIÓN

En esta sección explicaré dónde ha habido desvíos en cuanto a la planificación inicial mediante una tabla.

Sección	Inicial	Final
Investigación bibliográfica	20h	30h
Módulo procesamiento de imágenes	60h	70h
Módulo de búsqueda de esquinas	40h	25h
Módulo de transformada Geométrica	40h	30h
Módulo de extracción de respuestas alumno	40h	30h
Interfaz	20h	30h
Testeo aplicación	50h	80h
Documentación	30h	35h
Total	300h	330h

Como podemos apreciar ha habido un desvío de 30 horas aproximadamente respecto a la planificación inicial, esto se ha debido a problemas surgidos en diferentes partes del proyecto que ha provocado retrasos y que la aplicación se haya optado por hacerla en matlab para escritorio y no para móvil como inicialmente se había planteado. Ha habido un retraso especialmente en la fase de preprocesamiento de la imagen, que resulta ser la fundamental para que el resto de procesos de la aplicación funcionen correctamente. También ha habido un especial incremento de horas en la fase de testeo de la aplicación, ya que se ha tenido que probar muchas situaciones para asegurar un rendimiento medianamente fiable. Estos retrasos suelen ocurrir en aplicaciones del campo de la visión por computador ya que resulta realmente difícil asegurar fiabilidad alta debido a que intervienen diversos factores del mundo físico.

5 FUNCIONAMIENTO

La aplicación realizada, dadas dos imágenes de un mismo modelo de examen, una primera original sin llenar y otra segunda con las respuestas del alumno es capaz de encontrar la posición de las respuestas del alumno y calcular la nota a partir de una lista en formato "txt" con las diferentes respuestas correctas ingresadas por el profesor.



Fig. 1. Esquema del funcionamiento básico de la aplicación y sus diferentes módulos.

5.1 Funcionamiento básico de la interfaz

Para corregir un examen hay que seguir los pasos siguiente en el interfaz de la aplicación

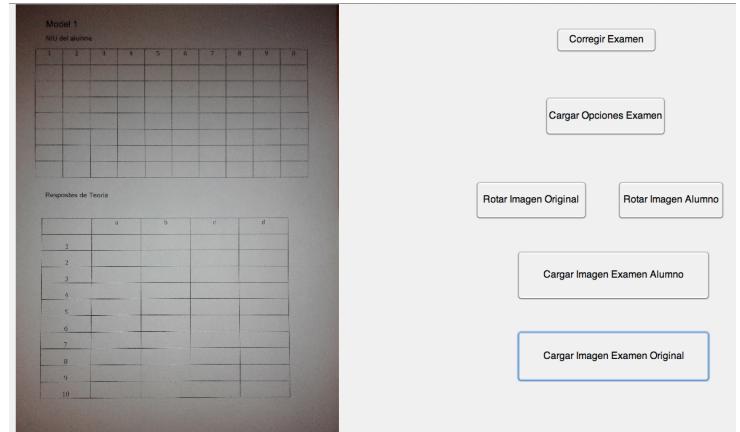


Fig. 2. interfaz de la aplicación

1. Pulsar botón Cargar Imagen Examen Original. Se visualizará en la parte izquierda la imagen del examen original, en el caso de estar horizontal apretar el botón Rotar Imagen Original.
2. Pulsar botón Cargar Imagen Examen Alumno .Se visualizará en la parte izquierda la imagen del examen del alumno, en el caso de estar horizontal apretar el botón Rotar Imagen Alumno.
3. El paso 1 solamente se tendrá que realizar una vez y en el caso de que la imagen original no cumpla los requisitos de inclinación se obligará al usuario a realizar otra fotografía más recta
4. Pulsar botón Cargar opciones examen donde se cargarán las opciones de las respuestas correctas del examen en el archivo "opciones.txt"
5. El paso 4 únicamente se tendrá que realizar una vez igual que en el paso 1, siempre y cuando no se cierre la aplicación.
6. Pulsar botón Corregir examen y aparecerá el examen a la izquierda binarizado y un mensaje con la nota del alumno

La interfaz es muy simple e intuitiva, pero estéticamente muy mejorable lo cual se propondrá como futuras mejoras.

5.2 Filtrado de imágenes

Para realizar el filtrado de las imágenes se utilizará un algoritmo llamado Niblack[7] el cual se basa en el principio de binarización local.

La idea básica es coger un umbral diferente para cada punto de la imagen.

A continuación explico el funcionamiento básico del algoritmo:

$$T(x,y) = m(x,y) + k \bullet s(x,y) \quad (0.1)$$

Donde $m(x,y)$ es la media local y $s(x,y)$ es la desviación estandar local. La vecindad ha de ser lo suficientemente pequeña para preservar los detalles, lo suficientemente grande para suprimir el ruido y como mínimo el radio de búsqueda debe ser el elemento de mayor grosor de la imagen. El valor k ajusta que parte de la frontera será asignada al objeto segmentado.

Una vez tenemos la imagen segmentada y binarizada aplicaremos un filtro por área, es decir eliminaremos todos aquellos elementos con un área menor a un determinado valor experimental para eliminar la presencia de ruido y asegurarnos que posteriormente podamos encontrar los corners correctamente.

5.3 Búsqueda de corners

Para la búsqueda de corners se utiliza un algoritmo clásico llamado algoritmo de Harris [x], el cual ha sido am-

pliamente utilizado en multitud de proyectos y ofrece muy buenos resultados. Fue propuesto el año 1988 y se basa en el análisis del tensor estructural 2D (matriz de segundas derivadas y matriz de segundo momento). En este proyecto se ha optado por el uso de este algoritmo porque presenta dos ventajas clave como son su invariancia a la rotación y su invariancia parcial a los cambios de intensidad.

El funcionamiento básico del algoritmo es el siguiente:

$$c(x,y) = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \quad (1.1)$$

I_x e I_y son los gradientes vertical y horizontal de la imagen. Definimos los autovalores λ_1 y λ_2 como los autovalores de la matriz de arriba, entonces se puede definir la función de autocorrelación R como:

$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2) \quad (1.2)$$

En el que K es un valor experimental.

En esta función R se pueden distinguir tres casos

1. La función tendrá un máximo si los dos autovalores son elevados, lo que significará que un desplazamiento en cualquier dirección provocará un incremento importante con lo qual se tratará de una esquina.
2. Si los autovalores son bajos, la función será casi nula, es decir, se tratará de una región que pertenece al mismo objeto.
3. Si un autovalor es bajo y el otro elevado significará que existe la presencia de un borde.

Pese a los buenos resultados que nos proporciona el algoritmo surge un problema y es que la imagen al ser binarizada no contiene líneas perfectamente rectas, es decir, que cada línea de las tablas de la imagen tiene un gran cantidad de corners ya que son irregulares, cosa que hace que el algoritmo encuentre muchos corners de los cuales tendremos que seleccionar los que se están buscando, que son los cuatro correspondientes a cada una de las tablas. Para hacer ésto se aplica el algoritmo de Harris, con un nombre de corners de 50000 lo cual nos asegurará el encuentro de todos los corners de la imagen.

Posteriormente a este proceso, de éstos 50000 tendremos que quedarnos con los 8 que nos interesan (4 de la primera tabla y 4 de la segunda)

Para lograr todo lo anterior, se partirá la imagen en dos, una imagen con la tabla superior y otra con la inferior respetando el tamaño original de la imagen para luego poder fusionarlas de nuevo y obtener las coordenadas que buscamos. Posteriormente se utilizará la fórmula de la distancia entre dos puntos que es la siguiente:

$$d(A,B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1.3)$$

Donde A es (x_1, y_1) y B es (x_2, y_2)

Lo que se hará es calcular la distancia de cada uno de los corners encontrados con respecto a las cuatro esquinas de la imagen (esquina superior izquierda, esquina superior derecha, esquina inferior izquierda y esquina inferior derecha) y nos quedaremos con los 4 valores que tengan la distancia más pequeña a estos 4 puntos y esos serán los corners exactos que buscamos.

Cabe destacar que esta solución forma parte de las técnicas algorítmicas llamadas voraces que se basan en seguir una determinada heurística (en este caso es la distancia más pequeña) y consiste en elegir la opción óptima en cada paso local con la esperanza de llegar a una solución general óptima.

Todo lo anterior presenta problemas ya que el algoritmo no tiene marcha atrás y no siempre encontraremos la mejor solución.

En este caso particular del proyecto, si en la fotografía no aparece solamente hoja de examen, es decir aparece en la imagen algún objeto diferente a la hoja de papel al binarizarse la imagen, se generará ruido en esa parte lo que hará que el algoritmo de búsqueda de corners falle.

Para este proyecto en particular se ha puesto como pre-requisito encuadrar perfectamente la imagen para obtener los resultados esperados. Posteriormente en el apartado de mejoras futuras se explicará el uso de otro método algorítmico que daría resultados más fiables en diferentes condiciones.

5.4 Transformación geométrica

Una vez se tienen los 8 corners localizados de cada una de las dos imágenes (la original y la del alumno), se trata de buscar aquella transformación geométrica que mayor resultados nos de, es decir que se aproxime más a la posición del examen original.

Para realizar ésto se utiliza una transformación geométrica llamada transformación afín.

Para poder encontrar la transformación geométrica que hay que hacer un proceso llamado "point matching" o mapeo de puntos que consiste en encontrar un número de puntos n de la imagen del alumno que correspondan a la imagen original.

En este caso usaremos los 8 corners de la imagen del alumno con los 8 corners de la imagen original.

Matemáticamente el proceso sería el siguiente :

$$P = \begin{bmatrix} x_0 & x_1 & \dots & x_{n-1} \\ y_0 & y_1 & \dots & y_{n-1} \\ 1 & 1 & \dots & 1 \end{bmatrix} = \begin{bmatrix} p_0 & p_1 & \dots & p_{n-1} \end{bmatrix}$$

$$Q = \begin{bmatrix} u_0 & u_1 & \dots & u_{n-1} \\ v_0 & v_1 & \dots & v_{n-1} \\ 1 & 1 & \dots & 1 \end{bmatrix} = \begin{bmatrix} q_0 & q_1 & \dots & q_{n-1} \end{bmatrix}$$

(1.4)

Aplicando la fórmula siguiente:

$$H = QP^T (PP^T)^{-1}$$

obtendremos la matriz de transformación H que estamos buscando que nos mapeará cada punto de la imagen del alumno con el de la imagen original.

En lenguaje matlab para realizar este cálculo utilizaremos dos funciones básicas como son:

- *Cp2tform(movingPoints,fixedPoints,transformtype)* donde movingPoints son las coordenadas x e y de los 8 puntos de la imagen del alumno , fixedPoints son las coordenadas x e y de la imagen original y transformtype es el tipo de transformación que se utilizará, que en nuestro caso será transformación afín. Para poder realizar este tipo de transformación se necesita como mínimo 3 puntos de cada imagen.
- *imwarp(A,tform)* donde A es la imagen que se quiere enderezar y tform es la matriz de transformación previamente calculada con la función *cp2tform*

5.5 Obtención de las respuestas del alumno

El correcto funcionamiento de este proceso depende de todos los anteriores y especialmente de la foto original del examen (la cual se hace únicamente una vez), que tiene que tener muy poca presencia de inclinación y poca perspectiva (casi nula).

En este proyecto sólo se podrá utilizar una plantilla en concreto que tiene varias limitaciones como son:

- Identificación sólo mediante NIU
- Número máximo de preguntas de 10
- Número máximo de opciones de 4
- Respuestas corresponderán a las filas
- Opciones corresponderán a las columnas

Estas restricciones han sido impuestas para asegurar un rendimiento fiable debido a la dificultad y a la falta de tiempo para realizar este proyecto.

En la sección de líneas futuras se explicarán posibles mejoras que puede tener este proyecto para no ser tan limitado a una plantilla.

El proceso es el siguiente:

1. Al tratarse de un modelo de examen que tiene NIU como identificación, para encontrar la altura de cada celda habrá que dividir la altura total de la tabla superior entre 8(7 dígitos del NIU mas una fila extra) .Para encontrar la anchura de la celda habrá que dividir la anchura total de la tabla superior entre 10 que son los posibles dígitos (1,2,3,4,5,6,7,8,9,0).
2. Leer número de respuestas del txt llamada "opciones.txt"
3. Leer número de opciones del mismo txt
4. Una vez procesados los datos se crearán dos matrices de las mismas dimensiones que las dos ta-

- blas, las cuales estarán llenas de ceros.
5. Finalmente habrá que buscar el centro de las celdas y mediante un rango de la anchura de la celda dividido entre tres y la altura de la celda dividido entre tres y mirar si en esa región hay un número de unos lógicos (es decir pixeles blancos que corresponderían a las respuestas del alumno).
 6. Para cada centro de celdas contabilizamos el número de unos lógicos que aparecen y lo anotamos en la matriz de ceros anteriormente creada (paso 4)
 7. Recorremos estas dos matrices y en las posiciones que no aparezcan ceros querrá decir que en esa celda existía una respuesta del alumno por lo tanto la mostramos como respuesta en la interfaz.

A continuación muestro los resultados de lo anterior mediante imágenes para mayor claridad.

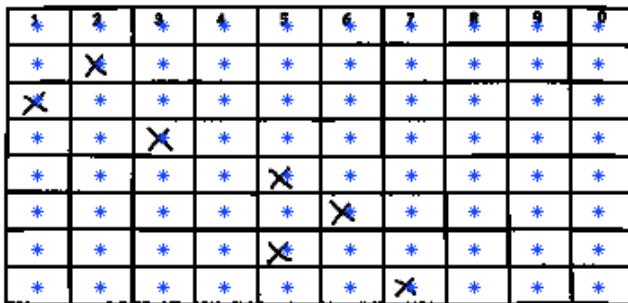


Fig. 1. Imagen con el centro de cada celda marcado en azul

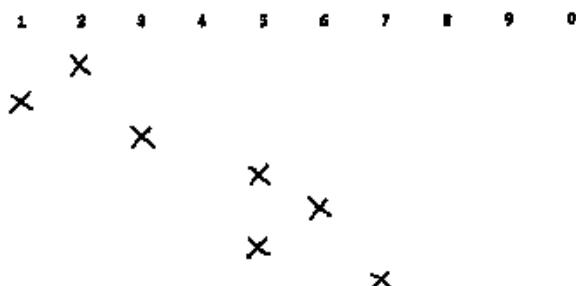


Fig. 2. Imagen sin líneas sobre la cual se extraerán las respuestas de los alumnos

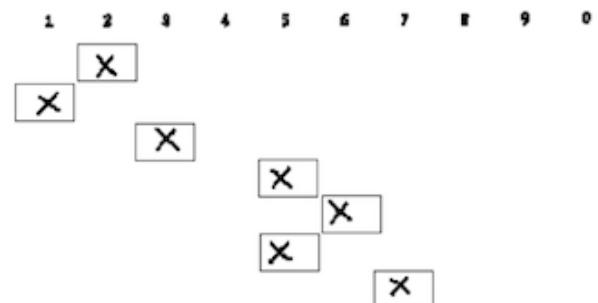


Fig. 3. Cada recuadro de la imagen significa el área de búsqueda de respuestas de cada celda. Toda respuesta que quede dentro de ese ámbito será contabilizada.

6 RESULTADOS

Primeramente mostraré varias imágenes de "buenas y malas prácticas":

1. Ejemplo de imagen bien enfocada y mal enfocada y su resultado al binarizarse.
2. Ejemplo de imagen con buena y mala iluminación y su resultado al binarizarse.
3. Ejemplo de fotografía con buena y mala inclinación.
4. Encuadre
5. Ejemplo de respuesta de alumno que binariza correctamente y ejemplo de respuesta que será propensa a fallos de la aplicación.

6.1.1 Enfoque

Esta parte es fundamental para el correcto funcionamiento de la aplicación .En este proyecto no se ha incorporado un módulo que mida el desenfoque de una imagen, cosa que ayudaría al usuario, ya que en el caso de realizar una foto desenfocada le avisaría obligándole a hacer otra fotografía.

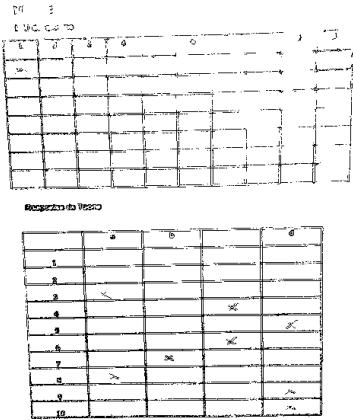


Fig.1. Ejemplo de imagen mal binarizada por presencia de desenfoque

6.1.2 Iluminación

Este factor es uno de los más importantes para el funcionamiento correcto de la aplicación, a su vez es uno de los más variables y que puede causar más problemas. Todas las fotografías deben hacerse con flash para evitar más problemas con la presencia de ruido ya que el uso del flash disminuye la presencia de ruido.

A continuación aparecerá un ejemplo de binarización de una fotografía con muy mala iluminación y otra con buena.

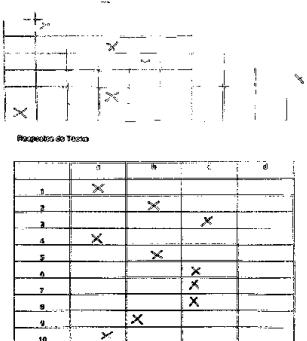


Fig. 2. Ejemplo de imagen mal binarizada por presencia de muy poca iluminación

Fig. 3. Ejemplo de imagen bien binarizada por presencia de buena iluminación

6.1.3 Inclinación

Este factor sólo se tendrá en cuenta en el examen del alumno, ya que el original tendrá que tener un grado de inclinación menor a 0.2. Ya que si el examen original supera estos grados de inclinación al realizar la búsqueda el centro de las celdas para poder extraer las respuestas no funcionará correctamente. Estos grados de inclinación son calculados de la siguiente forma

$$\text{puntoA} = (x_1, y_1)$$

$$\text{puntoB} = (x_2, y_2)$$

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad (1.5)$$

$$\text{inclinacion} = a \tan(m)$$

Donde m es la pendiente e inclinación como su propio nombre indica refleja la inclinación existente entre los dos puntos (A,B) y se expresará en grados.

Fig. 4. Ejemplo de imagen con buena inclinación.

Modelo 1 HuJ del alumno									
1	2	3	4	5	6	7	8	9	10
X									
	X								
		X							
			X						
				X					
					X				
						X			
X							X		

Respuestas de Teoria

	a	b	c	d
1	X			
2		X		
3			X	
4	X			
5		X		
6			X	
7			X	
8				X
9		X		
10	X			

Fig. 5. Ejemplo de imagen con mala inclinación.

A continuación se exponen unas tablas que reflejan los resultados de la aplicación con presencia de diferentes grados de inclinación. En este caso se han realizado 24 fotografías con cada uno de los tres modelos de móviles expuestos a continuación.

Modelo	Inclinación	Porcentaje
Iphone 5S	0 - 1	6/6 = 100 %
	1-1,5	6/6 = 100 %
	> 1,5	3/12 = 25%

Tabla 1. Como se puede apreciar existe un 100% de acierto con inclinaciones inferiores a 1,5 grados y un 25% en inclinaciones superiores a 1,5 grados

Modelo	Inclinación	Porcentaje
Samsung Galaxy Grand Duos	0 - 1	8/8 = 100 %
	1-1,5	8/8 = 100 %
	> 1,5	0/8 = 0%

Tabla 3. Como se puede apreciar existe un 100% de acierto con inclinaciones inferiores a 1,5 grados y un 0% en inclinaciones superiores a 1,5 grados

6.1.4 Encuadre

Este factor es uno de los más importantes ya que si la foto no está perfectamente encuadrada a la hora de ocupar el papel, al binarizarse la parte que no es papel se tomará como ruido y por lo tanto la búsqueda de corners fallará y como consecuencia fallarán todos los procesos posteriores (transformación geométrica y búsqueda de respuestas). Por lo tanto se ha puesto como prerequisito fundamental realizar un encuadre correcto. Cabe destacar que este prerequisito no se hubiera tenido que poner si la aplicación se hubiera realizado para móvil ya que se habría realizado un módulo de encuadre exacto del papel como utilizan muchas otras aplicaciones.

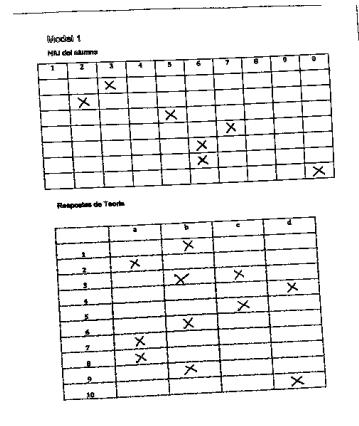


Fig. 6. Ejemplo de imagen con mal encuadre que implica presencia de ruido lateral

En la imagen anterior [6] se puede apreciar como la imagen al estar mal encuadrada todo aquello que no es papel al binarizarse se convierte en ruido lo que provocará malos resultados, por lo tanto como hemos dicho se trata de un factor primordial.

Modelo	Inclinación	Porcentaje
BQ Aquaris E5 HD	0 - 1	9/9 = 100 %
	1-1,5	3/3 = 100 %
	> 1,5	3/12 = 25%

Tabla 2. Como se puede apreciar existe un 100% de acierto con inclinaciones inferiores a 1,5 grados y un 25% en inclinaciones superiores a 1,5 grados

6.1.5 Respuestas alumno

Esta fase es fundamental también para obtener buenos resultados. Los alumnos a la hora de responder, si se quiere ofrecer un rendimiento fiable deberán responder con una x sin tocar las líneas de las celdas siempre cuento mas centrado mejor.

En las siguiente imágenes muestro un ejemplo claro:

	a	b
1		X
2	X	
3		X

Fig. 7. Ejemplo de imagen con buenas y malas marcas no binarizada

	a	b
1	*	X
2	X	
3		X

Fig. 8. Ejemplo de imagen con buenas y malas marcas binarizada

Como podemos apreciar a la hora de binarizar la imagen en la pregunta 3 opción B, la marca se une a la línea de abajo lo que implicará que a la hora de filtrar por área no se eliminan las marcas correctamente y por lo tanto esa respuesta se procesará mal. En cambio la pregunta 1 opción B, la respuesta está separada de la línea inferior y superior, cosa que favorecerá un correcto filtro por área y por consiguiente se podrá encontrar correctamente la respuesta.

A continuación se muestra una tabla con el porcentaje de acierto de 20 imágenes de respuestas de personas diferentes y el porcentaje de acierto de la aplicación.

Modelo	Número de fotos	Porcentaje acierto
BQ aquaris 5 HD	10	7/10 = 70%
Iphone 5S	10	9/10 = 90%

Tabla 4. Tabla que muestra el porcentaje de acierto en la extracción de las respuestas y del NIU de los alumnos.

Cabe destacar que de este porcentaje las fotografías que

fallan en la corrección son por dos causas como son exceso de inclinación y respuestas de los alumnos excesivamente cercanas a las celdas del examen. Ambos factores si se tienen en cuenta se obtienen muy buenos resultados.

En el dossier del tfg aparecen numerosas pruebas diferentes en muchas condiciones como iluminación general artificial, iluminación natural, iluminación artificial superior, iluminación artificial lateral y diferentes tipos de inclinación.

7 TRABAJOS FUTUROS

Existen varias partes que se pueden mejorar para incrementar el rendimiento y fiabilidad de la aplicación. Por ejemplo a la hora de detectar esquinas se podrían aplicar otras técnicas más seguras que los algoritmos Greedy utilizados como son Backtracking que nos ofrecerían la solución óptima global. En cuanto al módulo de transformación geométrica se podrían utilizar algoritmos de detección de puntos clave como son (SURF, SIFT etc) y posteriormente aplicar un algoritmo RANSAC para poder obtener la mejor de las transformaciones geométricas entre las dos imágenes.

También se podría incorporar un módulo de búsqueda de desenfoque en una imagen y en el caso de que la imagen estuviese desenfocada que avisará al usuario de volver a realizar la fotografía.

Finalmente esta aplicación se puede portar a plataformas móviles lo cual le da una mayor utilidad ya que se puede corregir y obtener la nota del examen en el momento de realizar la foto.

8 CONCLUSIONES

Podemos decir que la creación de un sistema corrector de exámenes tipo test mediante fotos es una tarea muy difícil de resolver ya que depende de muchos y variados factores físicos como son iluminación, enfoque de la imagen, encuadre de la misma y los diferentes tipos de respuestas de los alumnos.

Para asegurarnos de un funcionamiento correcto de la aplicación hemos puesto una serie de restricciones explícadas previamente para poder ofrecer una fiabilidad lo más elevada posible.

Pese a las dificultades del proyecto estoy satisfecho con los resultados finales ofrecidos con la escasez de tiempo disponible para realizar dicho proyecto y de los conocimientos que he adquirido en el área de la visión por computador y el procesamiento de imágenes.

AGRADECIMIENTOS

Quiero dar las gracias mi familia y amigos por apoyarme y ayudarme en todo momento así como a mi tutor Francisco Javier Sánchez Pujadas por ayudarme en la realización del proyecto.

BIBLIOGRAFIA

- [1] "Matlab" Accedido en noviembre y diciembre , 2014. Disponible:
<http://es.mathworks.com/help/images/index.html> Referència 2
- [2] R. Szeliski 2011 Computer Vision Algorithms and Applications ISBN 9781-84882-935-0
- [3] P. Klein Semptiembre 2013. Coding the Matrix: Linear Algebra through Applications to Computer Science ISBN 978-0615880990
- [4] "Electrical Engineering & Computer Science". Accedido en noviembre, 2014. Disponible:
<http://inside.mines.edu/~whoff/EENG512/lectures/>
- [5] C. Harris and M.J. Stephens. A combined corner and edge detector. In Alvey Vision Conference, pages 147–152, 1988.
- [6] Johannes Bauer, Niko Sünderhauf, Peter Protzel: "Comparing Several Implementations of two Recently Published Feature Detectors". In Proc. of the International Conference on Intelligent and Autonomous Systems (IAV), September 2007
- [7] Apunts visió per computador Grau Enginyeria Informàtica UAB.