

Corretor Automático de Provas Objetivas

Artur Waquil Campana
Instituto de Informática
Universidade Federal do Rio Grande do Sul
awcampana@inf.ufrgs.br

Nícolas Casagrande Duranti
Instituto de Informática
Universidade Federal do Rio Grande do Sul
ncduranti@inf.ufrgs.br

I. INTRODUÇÃO

A correção de provas objetivas consome consideravelmente o tempo de professores, especialmente do Ensino Médio. Por ser mecânica e repetitiva, essa é uma atividade com potencial de automatização. Neste trabalho, propomos uma aplicação que visa a auxiliar professores, realizando a correção a partir de imagens capturadas de grades de respostas.

II. DESCRIÇÃO DO PROBLEMA

A correção automática consiste em verificar em uma ou mais grades quantas respostas estão corretas em relação a um gabarito. Definimos a grade de respostas como uma tabela contendo na primeira linha um identificador de cada questão e, na primeira coluna, as alternativas de resposta. Uma resposta, por sua vez, é definida como uma marcação em uma célula da tabela que não pertença à primeira linha ou à primeira coluna. Uma marcação pode ser, por exemplo, um x desenhado na célula ou um rabisco preenchendo-a. Além disso, cada coluna da tabela deve apresentar uma, e apenas uma, resposta.

Considera-se que nas imagens as grades de respostas apresentam pouca rotação. Por fim, as fotografias devem ter sido capturadas com a câmera aproximadamente paralela à folha. Na Fig. 1 mostra-se uma imagem levemente distorcida, mas adequada à correção automática.

III. SOLUÇÃO PROPOSTA

Nossa solução para a correção automática foi implementada na linguagem Python e utiliza a biblioteca OpenCV. Primeiramente identificam-se as alternativas corretas numa grade de respostas de gabarito. Na sequência, corrige-se um número qualquer de outras grades de respostas.

A identificação das alternativas marcadas numa fotografia de grade de respostas é feita em duas etapas. Primeiro é necessário detectar os pontos que definem a tabela na imagem,

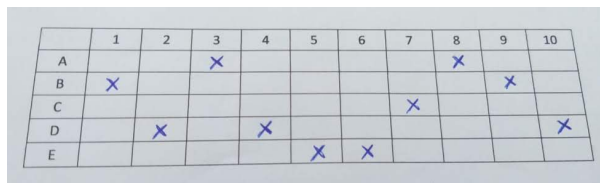


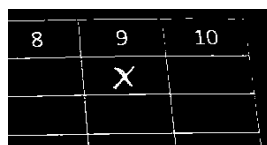
Fig. 1: Imagem com leve distorção

para em seguida avaliar quais respostas estão marcadas. Foram desenvolvidas duas abordagens diferentes para a detecção da tabela — uma baseada na Transformada de Hough e outra baseada no algoritmo proposto por Suzuki e Abe [1] — as quais são descritas nas subseções III-A e III-B.

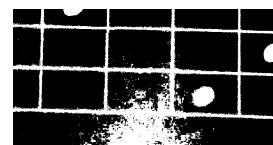
A avaliação de quais respostas foram marcadas na grade é feita da mesma maneira nas duas abordagens. Na imagem obtida após a detecção de bordas de Canny, percorrem-se todos os pixels de cada célula da tabela, contando quantos destes são brancos. Para cada coluna, a célula que possuir a maior quantidade de pixels brancos é a célula que foi marcada como resposta. Com essa informação, é possível comparar as respostas com o gabarito detectado, e determinar o número de acertos.

A. Solução baseada em Hough

A primeira estratégia de identificação da tabela baseia-se no uso da Transformada de Hough para detectar as linhas da grade de respostas. Primeiramente, utiliza-se o detector de bordas de Canny para se obterem os pontos a serem avaliados pela transformada. Utilizamos esse método em vez de fazer uma limiarização simples pois a variabilidade das sombras nas imagens não permite estabelecer um limiar único para todas elas, o que ocasiona bordas mal detectadas (Fig. 2a) e "manchas" de pontos não pertencentes à tabela (Fig. 2b). Em seguida, é aplicada a Transformada de Hough, selecionando picos com um mínimo de 150 votos, e obtém-se uma lista das linhas encontradas ordenada pela quantidade de votos recebidos. Após, baseando-se no parâmetro θ , são divididas as retas em dois grupos: as predominantemente horizontais e as predominantemente verticais.



(a) Bordas mal detectadas



(b) Mancha de pontos

Fig. 2: Efeitos da limiarização simples

Visto que são encontradas muitas retas (Fig. 3), são realizados dois processamentos para eliminar linhas redundantes (Fig. 4). Para todo par de retas não-paralelas pertencente a um

dos grupos, verifica-se se sua interseção acontece dentro dos limites da imagem — caso positivo, remove-se a linha cujo θ seja mais diferente em relação à mediana dos θ das 30% retas mais votadas no grupo. Em seguida, para retas de um mesmo grupo que sejam muito próximas (i.e. com d muito parecido) é removida aquela que tiver menos votos. Tendo o conjunto filtrado de retas, os pontos que definem a tabela são identificados calculando-se as interseções entre as retas dos diferentes grupos.

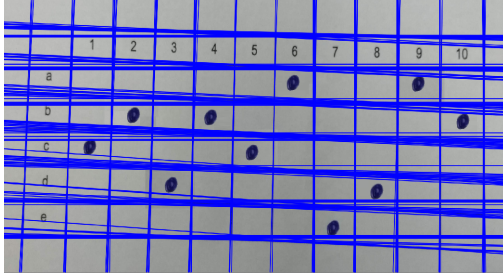


Fig. 3: Todas as linhas retornadas pela Transformada de Hough

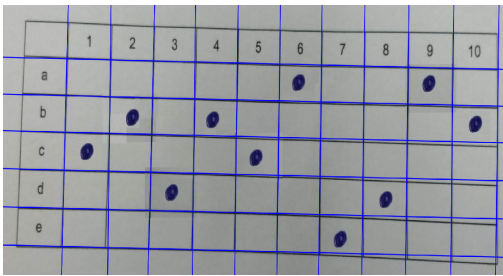


Fig. 4: Retas após a remoção de retas redundantes

B. Solução baseada em Suzuki

A outra estratégia se baseia no algoritmo de Suzuki-Abe de detecção de contornos. Primeiramente, é feita uma limiarização adaptativa na imagem da grade de respostas (pelo mesmo motivo que se usou Canny na primeira estratégia). Em seguida, para evitar a interferência de ruído na identificação dos contornos, são usadas as operações morfológicas de erosão e dilatação para detectar retas verticais e horizontais [2](Fig. 5). Após, somamos as duas imagens obtidas no passo anterior para gerar uma única imagem (Fig. 6).

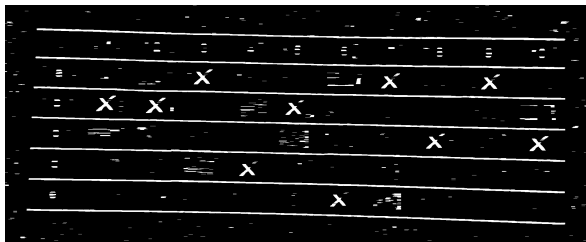


Fig. 5: Retas horizontais detectadas com as operações morfológicas

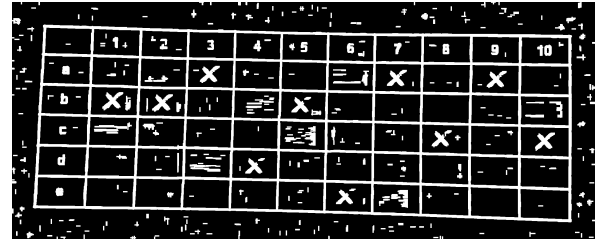


Fig. 6: Imagem utilizada como entrada para o algoritmo de Suzuki-Abe

O algoritmo de Suzuki-Abe detecta as bordas dos componentes conexos da imagem binária. É retornada pelo algoritmo uma lista de contornos, em que cada contorno é uma lista dos pixels que ele contém. Para novamente remover ruído da imagem, ignoramos os contornos muito pequenos ou que tenham lados muito distintos. Após, definimos *bounding boxes* em torno dos contornos, e eliminamos aqueles que sejam muito maiores ou menores que a moda dos tamanhos quantizados.

Para montar a tabela, é necessário ordenar os contornos. Como assumimos que as imagens são bem-comportadas, é possível determinar o número de linhas (e, conseqüentemente, o de colunas) selecionando o contorno mais à esquerda c e contando quantos contornos têm seu canto esquerdo mais à esquerda que o centro de c . Sabendo as dimensões da tabela pode-se particionar o conjunto de contornos em colunas, e então ordenar cada coluna pela altura do canto esquerdo da *bounding box* de cada contorno. Na Fig. 7 mostram-se os contornos detectados com esta abordagem.

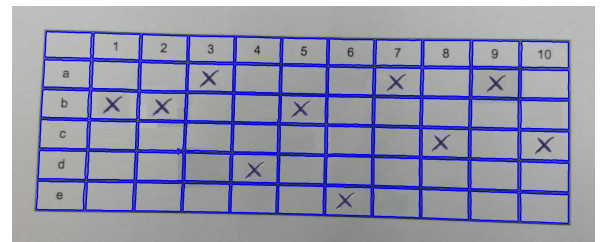


Fig. 7: Contornos detectados com a segunda abordagem

IV. RESULTADOS EXPERIMENTAIS

A solução baseada na Transformada de Hough mostrou-se muito promissora, gerando linhas que representavam a grade de respostas quase perfeitamente (Fig. 8 e Fig. 9). Entretanto, para algumas imagens, esta abordagem não funcionou muito bem. Por exemplo, quando as linhas verticais da tabela não são muito grandes, as retas que se ajustam a essas linhas não recebem uma quantidade suficiente de votos (Fig. 10). Ao se selecionarem picos com um número menor de votos, muitas outras linhas redundantes são identificadas (principalmente horizontais), as quais influenciam negativamente no resultado do processo de filtragem das retas.

Outro caso em que este procedimento não se mostrou adequado é quando a tabela é deformada em função da

captura da imagem, de modo que linhas originalmente paralelas não apresentam mais esta característica na foto gerada. Em decorrência disso, a estratégia de utilizar a mediana dos ângulos para eliminar linhas redundantes pode preservar retas que não são as mais adequadas (Fig. 9).

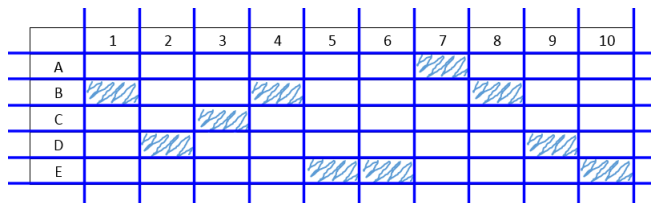


Fig. 8: Resultado utilizando a abordagem com Hough em um print de uma grade de respostas

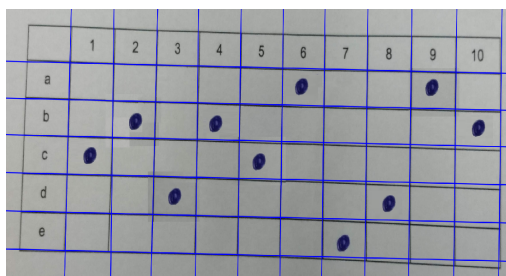


Fig. 9: Resultado utilizando a abordagem com Hough em uma foto de uma grade de respostas. Nas retas inferiores, é possível perceber o efeito da deformação da tabela na filtragem das linhas



Fig. 10: Exemplo utilizando a abordagem com Hough em que algumas linhas não foram identificadas.

Enquanto a estratégia com Hough suporta rotações de até 45°, a segunda abordagem funcionou bem para imagens com até 10° de rotação — acima disso não é possível detectar retas verticais e horizontais com as operações morfológicas. A montagem dos contornos é robusta a ruído na imagem, e funciona bem exceto para células da tabela que estejam completamente preenchidas (Fig. 11).

V. CONCLUSÕES

Com os resultados obtidos, podemos observar que a utilização da Transformada de Hough para a resolução deste problema apresenta pontos muito positivos. Além de gerar bons resultados, esta abordagem permite uma maior quantidade de rotação da grade nas imagens de entrada. Ainda, com esta técnica o algoritmo de identificação da tabela é muito mais simples e intuitivo, uma vez que conhecemos todas as linhas que a compõem, bem como suas localizações.

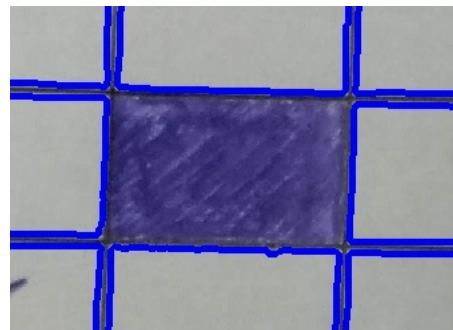


Fig. 11: Contorno não detectado devido a célula preenchida

Em contrapartida, também foi possível verificar que para algumas imagens a estratégia utilizada para a abordagem com Hough não foi muito bem-sucedida. Entretanto, acreditamos que isto se deve às escolhas de implementação e que, com métodos diferentes, estes problemas poderiam ser resolvidos.

A resolução utilizando o algoritmo de Suzuki-Abe, por sua vez, também gerou resultados muito bons. A identificação de cada célula individualmente possibilitou uma representação bastante precisa das grades de resposta. Contudo, uma vez que os contornos são estruturas individuais, o processo de identificação e “montagem” da tabela se torna menos intuitivo. Além disso, para que ele seja executado corretamente, é necessário restringir mais a rotação que a grade pode apresentar na imagem de entrada.

REFERÊNCIAS

- [1] S. Suzuki and K. Abe, Topological Structural Analysis of Digitized Binary Images by Border Following, CVGIP 30 1, pp 32-46 (1985)
- [2] “Extract horizontal and vertical lines by using morphological operations,” OpenCV. [Online]. Available: https://docs.opencv.org/3.4/dd7/tutorial_morph_lines_detection.html. [Accessed: 10-Dec-2019].