A SWT para detectar RBNs em corridas de rua

Gabriel Martins de Miranda
Universidade de Brasília
www.unb.br
gabrielmirandat@hotmail.com

Abstract

Automatizar o processo de reconhecimento de RBNs em corridas de rua ainda é um processo não resolvido. Grande parte do tempo gasto por empresas de fotografias está em etiquetar estas imagens, feito hoje manualmente. Apresentamos o poder de uma SWT diferenciada capaz de detectar com precisão larguras de traçados em imagens com caracteres de bordas finas (tags de corredores em corridas de rua). Do algoritmo temos um ponto de partida para poupar estas pessoas deste trabalho manual cansativo e maçante.

1. Introdução

Fotógrafos (ou empresas) que lidam com corridas de rua possuem a tarefa de etiquetar suas imagens para permitir acesso rápido a um determinado corredor. Automatizar este processo poderia poupar horas e horas de pobres pessoas que trabalham de forma maçante num movimento estressante e repetitivo. Autores como Boris Epshtein *et al*. proporam um brilhante ponto de partida em [1] para solucionar, de forma geral, problemas deste tipo, ou melhor, qualquer tipo de problema que envolva caracteres em imagens com cenários naturais. Apesar deste grandioso ponto de partida, ainda hoje não é uma questão completamente resolvida. Em vez de propor uma solução geral, nós pensamos de forma segmentada, ou seja, dirigir o que foi proposto em [1] para solucionar problemas deste tipo. Usaremos o termo proposto por BenAmi *et al*. em seu trabalho descrito em [2], Racing Bib Number Recognition (RBN) para nos dirigirmos às etiquetas utilizadas pelos corredores, cujo propósito primeiro do artigo será detectá-las no maior número possível destas fotografias. Observando imagens de corrida de rua, pudemos concluir algumas situações que frequentemente ocorrem e daí formular nosso algoritmo com base nestes fatos: a RBN pode possuir algarismos falhados. As magnitudes das larguras dos algarismos numa RBN não são elevadas. Há grandes variações de iluminação presente numa mesma RBN. As RBNs podem estar dispostas em várias orientações. Podem haver RBNs de apenas 2 algarismos.

Diferente de [2], que propõe encontrar possíveis regiões em que há maior probabilidade de se encontrar uma RBN para só então aplicar a SWT, nosso algoritmo é focado em direcionar a SWT para ganhar precisão e velocidade de forma inovadora.

2. Metodologia

Para lidar com o problema de reconhecer RBNs e tentar alcançar o estado da arte, as seguintes decisões foram tomadas:

1) Threshold para o detector de bordas

Os limites para o algoritmo de Canny, proposto em [3], passaram de 175 para 300 no limite inferior, e de 320 para 600 no limite superior. A mudança possui o objetivo principal de remover o ruído característico de RBNs falhadas. Os valores SW nos algarismos com muitos detalhes/ruídos característicos destas falhas acabam tornando-se muito variados o que ocasiona o descarte destes pelo processo. Como as RBNs possuem fundos bem destacados, como preto no branco, este mudança mantém apenas bordas relevantes.

Figure 1: Observe esta RBN. Em tags de corridas de rua é normal este tipo de falhas.

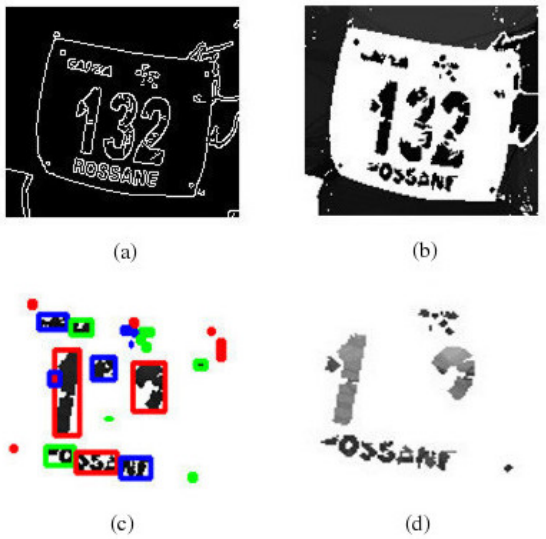

Figure 2: (a) Após detecção de bordas, com thresholds entre 175 e 320. Perceba a quantidade de ruído presente dentro dos algarismos (b) A transformada SW aplicada em (a). As falhas atrapalham consideravelmente o resultado da transformada. (c) Após descarte das BBs. Podemos ver que pedacos do algarismo foram descartados já na etapa anterior. (d) Resultado final, após descarte de BBs e chaining. Veja que grande parte do algarismo foi descartado.

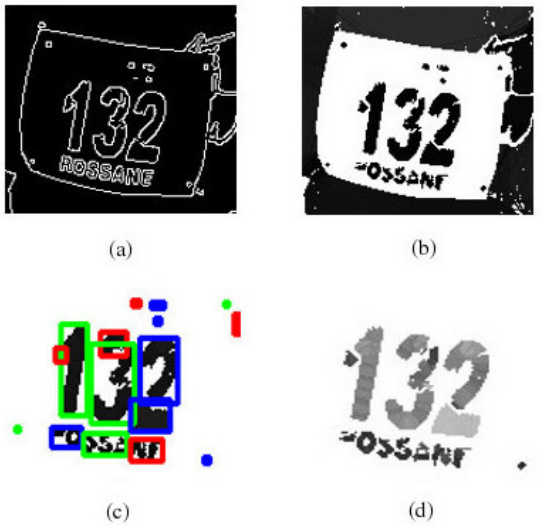

Figure 3: (a) Utilizando a mesma imagem, porém com os thresholds setados para 300 e 600. Observe que grande parte do ruído que estava presente dentro dos algarismos desapareceu. (b) A transformada SW aplicada em (a). Uma continuidade maior dos algarismos é obtida. (c) Mesmo após o descarte das BBs, vemos que o algarismo como um todo é mantido. (d) Resultado final, após descarte de BBs e chaining. Grande parte da RBN foi mantida.

2) Precisão multiplicada pelas imagens gradiente.

A precisão proposta, por ter um valor relativo elevado, adiciona elementos desnecessários que posteriormente serão utilizados pela SWT como ponto de partida, podendo ocasionar ruídos próximos às RBNs. O que fizemos foi diminuir esta constante de 0.5 para 0.2.

Figure 4: Uma RBN comum.

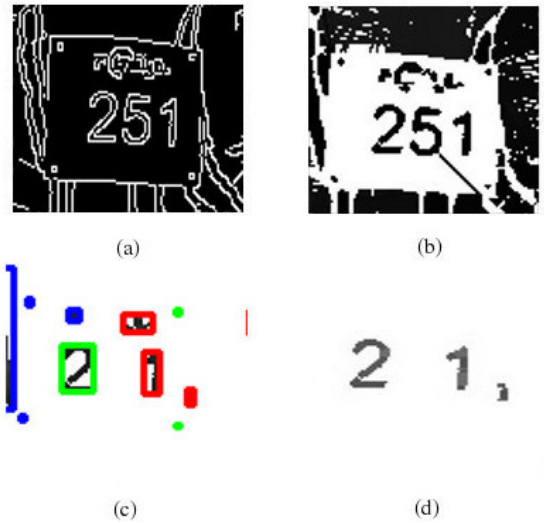

Figure 5: (a) Após detecção de bordas com Canny. (b) A transformada SW aplicada em (a). Observe que um falso valor de tracado surge devido à constante de 0.5. (c) Após descarte das BBs. Vemos que o algarismo que apresenta aquela falso tracado foi descartado. (d) Resultado final, após descarte de BBs e chaining.

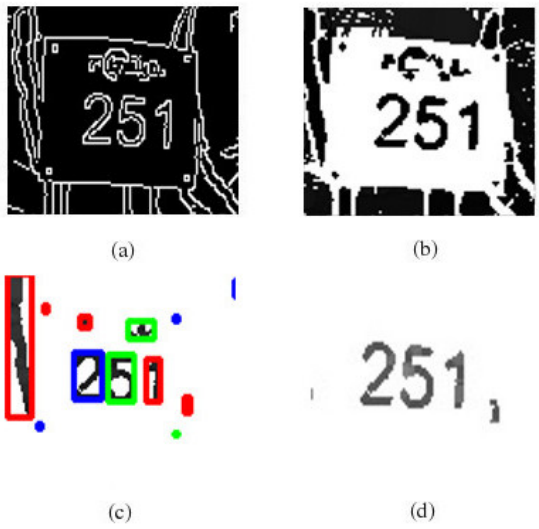

Figure 6: (a) Bordas através de Canny. (b) A transformada SW aplicada em (a). Observe que aquele falso tracado desaparece quando utilizamos a constante de 0.2. (c) Mesmo após o descarte das BBs, vemos que todos os algarismo da RBN sao mantidos. (d) Resultado final, bastante satisfatório.

3) Distância das cores das CCs

Numa mesma RBN é comum que exista grandes variações de luz, já que grande parte das corridas ocorre à luz do dia e o local da tag reflete bastante pois frequentemente possui fundo branco. Pode acontecer também que parte da tag fique sombreada, dificultando o reconhecimento se a restrição de cores para o chaining for muito elevado. Com isto, alteramos este valor de 1600 para 5000.

Figure 7: Uma RBN comum. Perceba que o algarismo 5 se encontra mais sombreado que os outros.

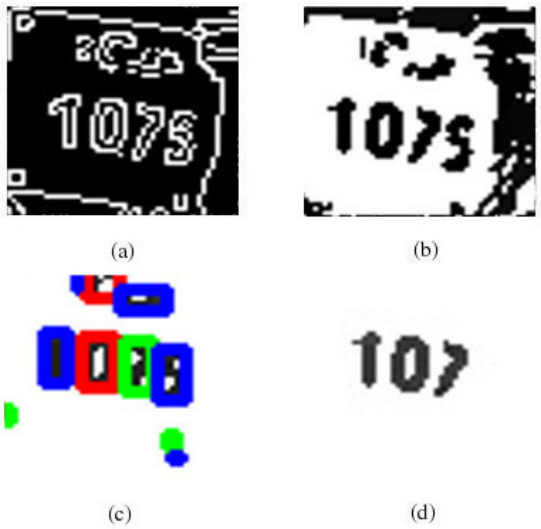

Figure 8: (a) Após detecção de bordas com Canny. (b) A transformada SW aplicada em (a). (c) Após descarte das BBs. (d) Resultado final. Podemos ver aqui que o algarismo 5 foi descartado do número encontrado. Isto acontece pois este algarismo está mais sombreado que os outros da mesma RBN.

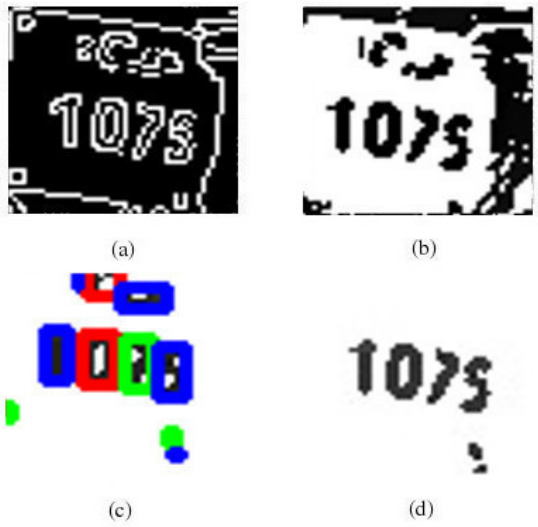

Figure 9: (a) Após detecção de bordas com Canny. (b) A transformada SW aplicada em (a). (c) Após descarte das BBs. (d) Resultado final, após descarte de BBs e chaining. Aqui, gracas a suavizacão das distâncias de cores, foi possível identificar o algarismo 5 como algarismo pertencente à RBN como um todo.

4) Ângulo entre CCs para chaining

Durante a corrida, muitas RBNs podem rotacionar no corpo do corredor. Com isto, o ângulo entre os algarismos

pode variar bastante. Pensando nisto, mudamos a restrição de que para se formar um chaining entre as BBs seria permitido no máximo um ângulo de 30 graus e passamos este ângulo para 60 graus.

Figure 10: Uma RBN comum. Perceba que ela se encontra rotacionada.

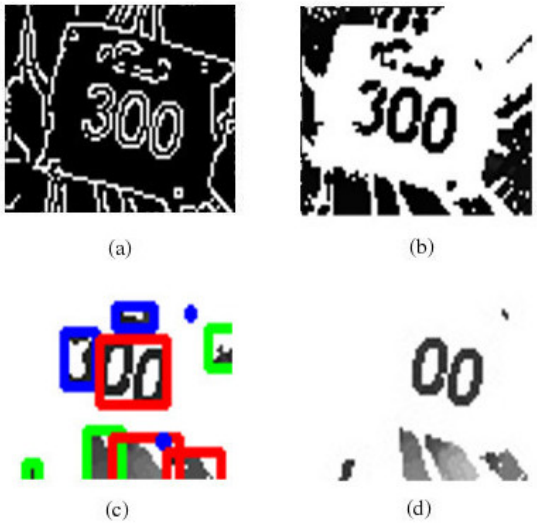

Figure 11: (a) Após detecção de bordas com Canny. (b) A transformada SW aplicada em (a). (c) Após descarte das BBs. (d) Resultado final. Como o algarismo 3 possui um certo ângulo com os algarismo zeros, ele foi descartado.

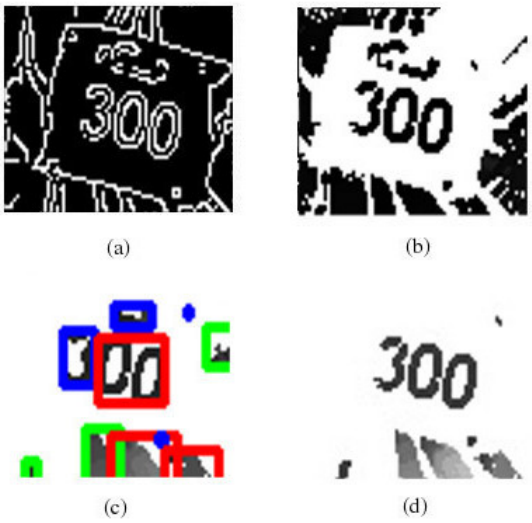

Figure 12: (a) Após detecção de bordas com Canny. (b) A transformada SW aplicada em (a). (c) Após descarte das BBs. (d) Resultado final, após descarte de BBs e chaining. Por permitir ângulos de até 60 graus, consegue-se unir o algarismo 3 á sua RBN.

5) Número mínimo de BBs para que se forme uma chain

RBNs frequentemente possuem de 2 a 5 algarismos. Para tentar identificar a maioria das tags, mudamos a restrição para que seja necessário um mínimo de duas BBs para que se forme uma chain.

Figure 13: Uma RBN comum. Note que ela só é formada por dois algarismos.

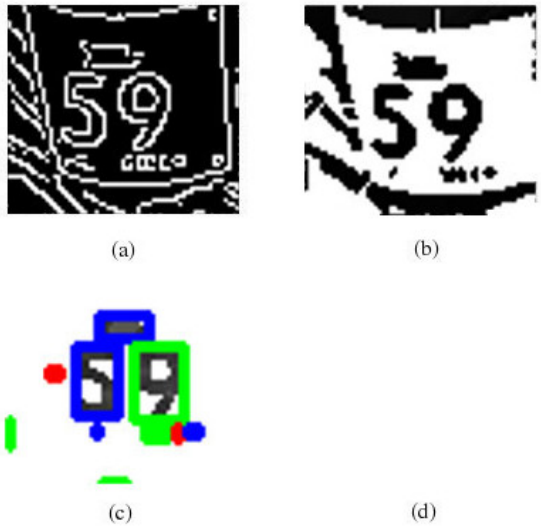

Figure 14: (a) Após detecção de bordas com Canny. (b) A transformada SW aplicada em (a). (c) Após descarte das BBs. (d) Resultado final. Nada aparece, já que não aconteceu chaining.

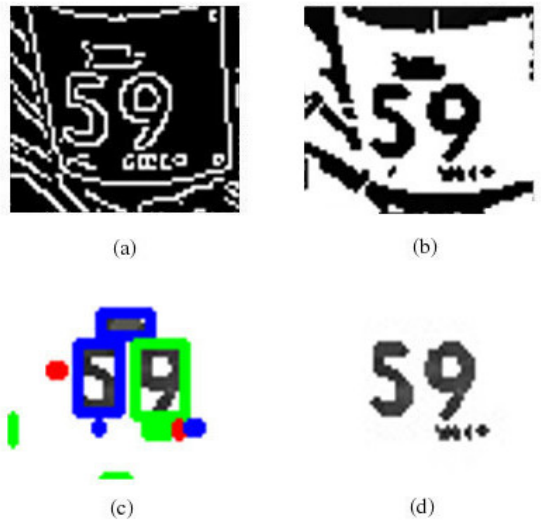

Figure 15: (a) Após detecção de bordas com Canny. (b) A transformada SW aplicada em (a). (c) Após descarte das BBs. (d) Resultado final, após descarte de BBs e chaining. Por permitir um mínimo de duas BBs, o chaining ocorre e a RBN não é descartada.

6) Valor mediano para o raio na segunda passada

Aqui escolhemos um valor menor que a mediana como SW para compor o raio. Isto homogeniza os valores dos traçados (característica também adquirida com a suavização na detecção de bordas) num mesmo algarismo e assim dificulta que ele seja rejeitado. Em vez de usarmos a mediana como valor SW de todo o raio, utilizamos o valor tamanho_do_vetor/12. Isto faz com que um valor SW menor em magnitude seja escolhido como o SW de todo o raio. Esta mudança é interessante quando analizamos o algarismo 4. Por ter um cruzamento perpendicular, possui tanto larguras horizontais quanto verticais identificadas pela SWT original. O que acontece é que a segunda passada do algoritmo para homogeneizar estes valores encontra dificuldades no processo, e frequentemente este algarismo é rejeitado. Observe as imagens que se seguem:

Figure 16: Algarismos de 0 a 9.

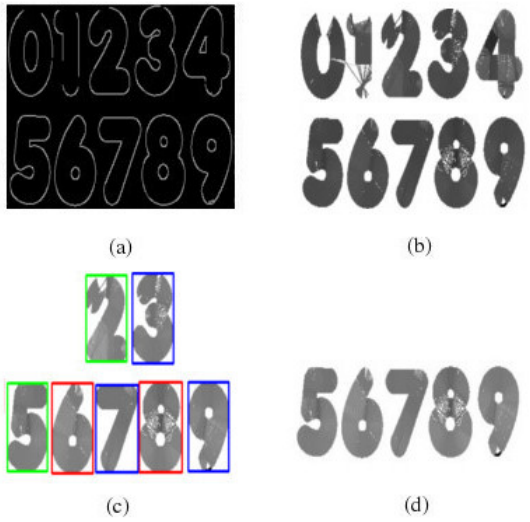

Figure 17: (a) Após detecção de bordas com Canny. (b) A transformada SW aplicada em (a). Note que algumas descontinuidades ocorreram nos algarismos 0 e 1 devido a falhas nas bordas. Porém, o algarismo 4 está em perfeitas condicoes. (c) Após descarte das BBs. O algarismo 4, mesmo estando em boas condicoes, foi rejeitado. Isto acontece devido à grandes diferencas dos valores SW nos raios que compõem o 4. (d) Resultado final. O 2 e o 3 foram rejeitados pois inicialmente não havia formacão de chain com apenas dois algarismos.

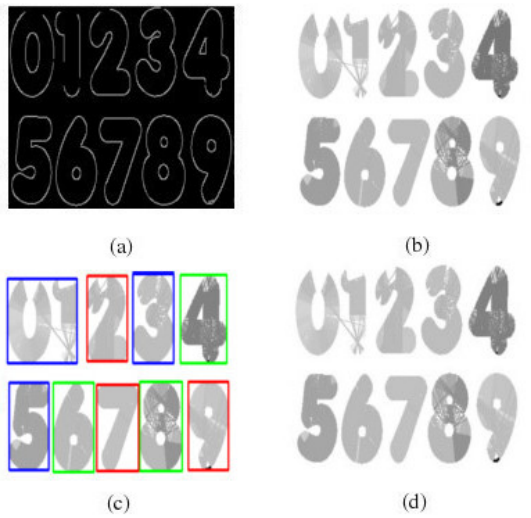

Figure 18: (a) Após detecção de bordas com Canny. (b) A transformada SW aplicada em (a). Perceba que agora as cores no algarismo 4 estão distribuídas de forma mais homogênea. (c) Após descarte das BBs. Nenhum algarismo foi rejeitado no processo de formacão das BBs. Note que devido ao problema na detecção das bordas, os algarismos 0 e 1 ficaram unidos em uma só BB. (d) Resultado final, após descarte de BBs e chaining. Todos os algarismos foram mantidos.

Vejamos na prática:

Figure 19: Uma RBN com algarismos 4.

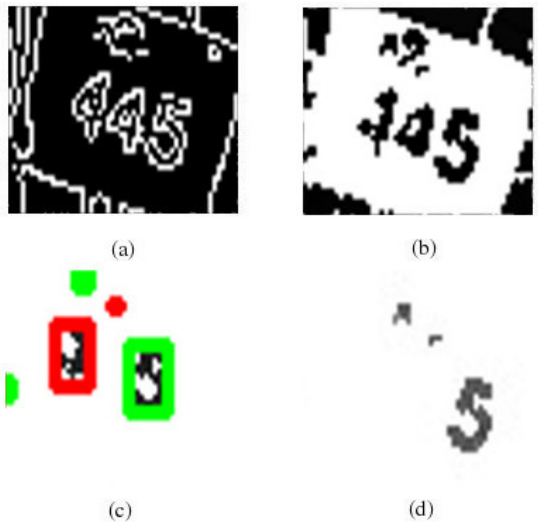

Figure 20: (a) Após detecção de bordas com Canny. (b) A transformada SW aplicada em (a). (c) Após descarte das BBs. Um dos algarismos 4 foi descartado. (d) Resultado final. Houve descarte de grande parte da tag.

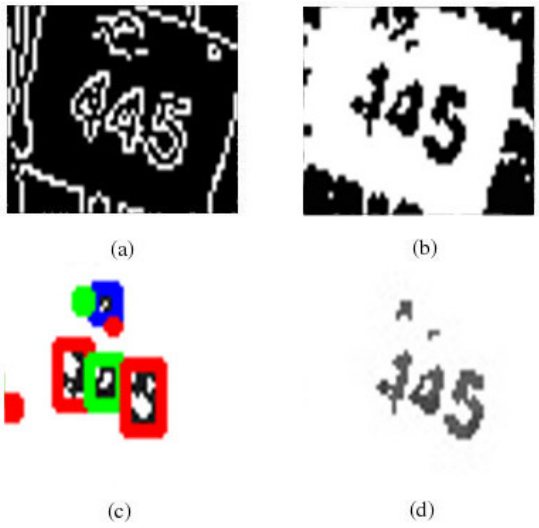

Figure 21: (a) Após detecção de bordas com Canny. (b) A transformada SW aplicada em (a). Note que não se consegue perceber diferença com o caso em que se usa a mediana. (c) Após descarte das BBs. Nenhum dos algarismos foi descartado. (d) Resultado final, após descarte de BBs e chaining.

7) Limitando o tamanho máximo dos traçados da SWT

Os resultados apresentados na SWT original apresentam larguras de qualquer tamanho, dispostas de qualquer forma. Acontece que os algarismos presentes numa RBN tem um

tamanho de largura do traçado extremamente limitado. Se considerarmos que uma RBN esteja disposta no peito de um corredor de forma que uma imagem tirada contenha o rosto dele na parte superior e a tag na inferior, de forma totalmente preenchida, podemos estimar que a largura dos algarismos não passam de 30 pixels. Diante disto, cortamos do resultado da SWT larguras acima de 30 pixels, e com isto obtivemos resultados interessantíssimos. Além de velocidade e remoção de ruído, conseguimos detectar em vários casos em que larguras não interessantes tornavam-se ruído e impediam a definição de BBs consistentes. Observe um exemplo:

Figure 22: Uma imagem de uma corrida.

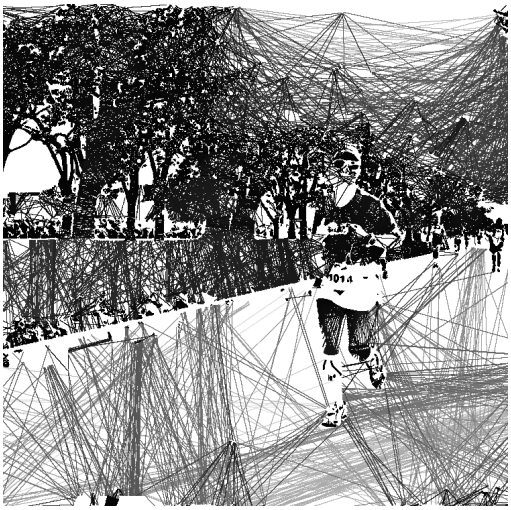

Figure 23: A SWT original aplicada na imagem. Observe a quantidade de informacão que não nos é interessante.

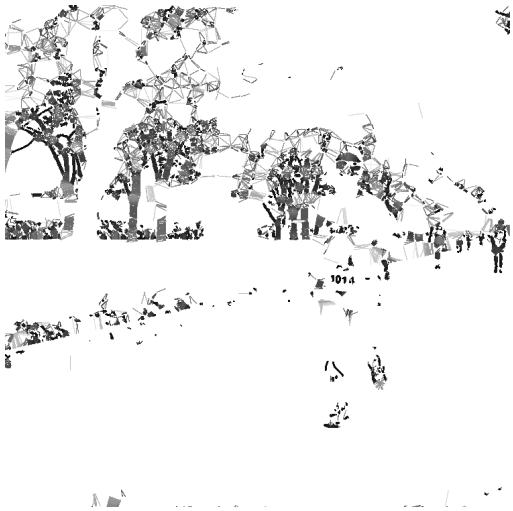

Figure 24: A SWT com restricão nos tracados. O que resta são apenas raios relevantes para a detecção de RBNs.

Olhando mais de perto, temos:

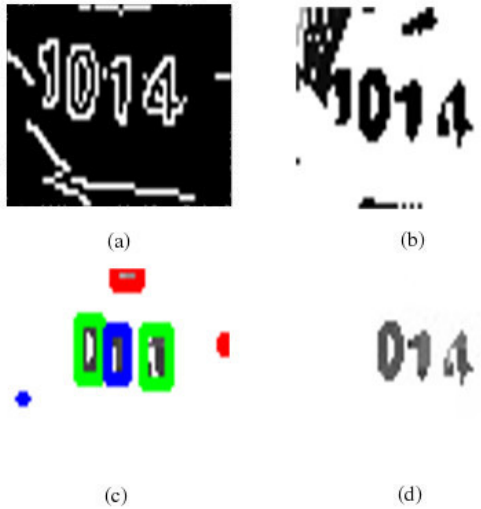

Figure 25: (a) Após detecção de bordas com Canny. (b) A transformada SW aplicada em (a). Observe a proximidade de raios irrelevantes com os algarismos da RBN. (c) Após descarte das BBs. O algarismo 1 foi rejeitado pois não foi possível criar uma BB que o envolvesse. (d) Resultado final, com a tag incompleta.

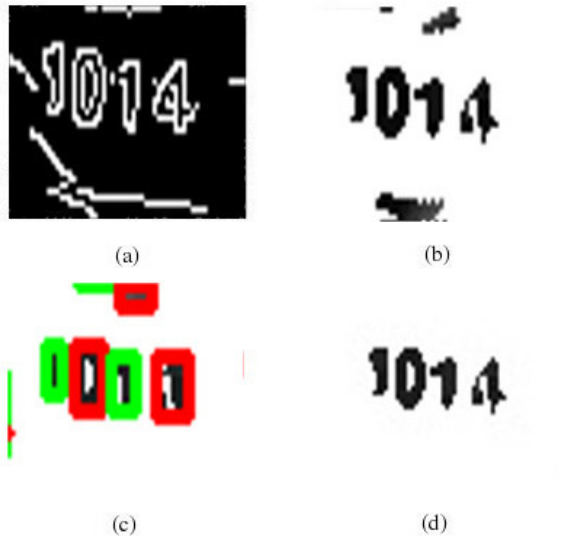

Figure 26: (a) Após detecção de bordas com Canny. (b) A transformada SW aplicada em (a). O raio irrelevante sumiu devido à restrição da largura máxima. (c) Após descarte das BBs. Vemos que aqui nenhum dos algarismos da tag é rejeitado. (d) Resultado final, após descarte de BBs e chaining. Nada é perdido e obtemos um resultado satisfatório.

3. Resultados

Num banco de 132 imagens de uma mesma corrida, obtivemos reconhecimento de todas as RBNs em 87.93% das imagens, contra 65.52% utilizando a SWT original. Das 14 imagens não reconhecidas, tinhamos imagens borradas e RBNs de tamanho muito pequeno. 16 das imagens eram irreconhecíveis devido à oclusão de parte da RBN ou não presença de nenhuma RBN.

Num banco de 125 imagens de corridas diferentes, obtivemos o resultado de 75.60% , contra 47.15% utilizando a SWT original. Das 30 não reconhecidas, uma estava relativamente borrada, 9 possuíam grande mudança de iluminação na RBN e as outras 20 possuiam RBN com pouco contraste com o fundo.

O desempenho também melhorou bastante. Contabilizando o tempo total para rodar ambos os algoritmos no banco de imagens completo com 257 imagens, nossa abordagem levou 4 minutos e 40 segundos contra 20 minutos e 48 segundos com a SWT original.

4. Conclusão

Neste trabalho mostramos como utilizar a Stroke Width Transform numa abordagem em que não se possui interesse em caracteres de tamanho de largura elevado, no caso, em RBNs de corridas de rua. Através da restrição do tamanho máximo de largura, conseguimos melhorar os resultados da SWT original, assim como aumentar a velocidade do algoritmo. Das imagens não reconhecidas, tinhamos imagens borradas, tags de tamanho muito pequeno e baixo contraste dos algarismos com o fundo. É importante salientar que este último torna-se um problema maior devido á suavização dos filtros de borda e da constante de precisão multiplicada pelas imagens gradiente. Além disto, apesar de termos minimizado a quantidade de ruído, ele surgiu de outras formas dadas as modificações, constituindo um viés do algoritmo. Sugerimos que trabalhos futuros se empenhem em corrigir estes problemas.

References

[1] E. O. B. Epshtein and Y. Wexler. Detecting text in natural scenes with stroke width transform. pages 2963–2970, 2010.

[2] B. T. Ben-Ami, I. and S. Avidan. Racing bib numbers recognition. pages 19.1–19.10, 2012.

[3] J. Canny. A computational approach to edge detection. PAMI-8:679 – 698, 1986.

	Banco 1 - 132 imagens				Banco 2 - 125 imagens				Banco total - 257 imagens
	Identificados	Não identificados	Irreconhecíveis	Acerto	Identificados	Não identificados	Irreconhecíveis	Acerto	Tempo gasto
Nosso algoritmo	102	14	16	87.93%	93	30	2	75.60%	4 min 40 seg
SWT original	76	40	16	65.52%	58	65	2	47.15%	20 min 48 seg

Table 1: Comparando nosso algoritmo com a SWT original.