Título da Pesquisa: Desenvolvimento de um sistema de rastreamento ocular para mensurar o desempenho de leitura.

Palavras-chave: eyetracker, rastreamento ocular, desempenho de leitura, visão computacional
Campus: Congonhas **Tipo de Bolsa**: PIBITI **Financiador**: CNPq

Bolsista (as): Monique Morais do Vale; Sabrina Moreira Costa
Professor Orientador: Fabrício Carvalho Soares

Área de Conhecimento: Educação / Avaliação da Aprendizagem ; Ciência da Computação / Processamento Gráfico (Graphics)

Resumo:

Entre os diversos fatores que representam o desenvolvimento escolar de um indivíduo, um dos mais importantes é o desempenho de leitura, pois é a base para novos aprendizados. Este trabalho tem como objetivo descrever o desenvolvimento inicial de um sistema de rastreamento ocular, de baixo custo, com o objetivo de mensurar o desempenho de leitura. Espera-se que ao final do desenvolvimento, durante a leitura de um texto pré-selecionado, será realizado um vídeo do aluno, com foco em seus olhos, que será processado para mensurar a velocidade de leitura, a quantidade de fixação, o tempo médio de fixação, o número de regressões, entre outros fatores. O software de processamento está sendo desenvolvido na linguagem C++ em conjunto com a biblioteca de processamento de imagens OpenCVpor permitir trabalhar em alto nível de forma rápida, também sendo possível empregar códigos que exijam desempenho em tempo real. O algoritmo criado foi capaz de reconhecer o objeto a ser rastreado, podendo mover dinamicamente. O objeto detectado é determinado por um retângulo ou por um círculo atualizando sua posição de acordo com sua movimentação, sendo esta posição armazenada para futura correlação com o texto exibido na tela de modo a obter os parâmetros desejados.

INTRODUÇÃO:

Os problemas visuais respondem por grande parcela de evasão e repetência escolar, pelo desajuste individual no trabalho e por grandes limitações na qualidade de vida, mesmo quando não se trata, ainda, de cegueira [4].

Os dados epidemiológicos disponíveis para o Brasil, segundo o Conselho Brasileiro de oftalmologia - CBO, mostram que 30% das crianças em idade escolar e 100% dos adultos com mais de 40 anos apresentam problemas de refração que interferem em seu desempenho diário e, consequentemente, na sua autoestima, na sua inserção social e em sua qualidade de vida [3].

Entre os diversos fatores que representam o desenvolvimento escolar de um indivíduo, um dos mais importantes é o desempenho de leitura, pois é a base para novos aprendizados. Com isso verificou-se a necessidade de desenvolver um sistema, devido a grande carência de equipamentos em alguns exames de triagem visual, onde os resultados obtidos após a devida análise serão armazenados em um banco de dados.

Pode-se realizar o exame de desempenho de leitura, cronometrando-se o tempo de leitura de um texto padrão ou seguindo e detectando o olho durante a leitura de um texto padrão, utilizando-se um equipamento chamado eyetracker capaz de mostrar a posição e a movimentação do olho humano. O eyetracker é um método viável de análise do movimento ocular, utilizado durante décadas para estudar o comportamento humano.

A área do conhecimento que atua nesse sentido é chamada de Biometria, onde as características físicas e comportamentais dos seres vivos podem ser medidas, tratadas e estudadas estatisticamente. Estas podem ser utilizadas como chave ou identificador pessoal para cada ser humano por serem únicas. Para se fazer medições diretamente nos corpos dos seres vivos, através da análise de imagens são utilizados instrumentos e ferramentas de máquinas e sistemas computacionais [2].

Como uma das mais salientes características da face humana, os olhos representam uma importante função na análise automática de expressões faciais, pois apresentam alguns pontos fiduciais bastante estáveis em comparação com outras características faciais. Os nossos olhos fazem uma série de fixações ou estabilizam espacialmente entre 200-300 milissegundos quando focam uma área particular, e os movimentos rápidos do olho ocorrem entre fixações de 40-50 milissegundos [1].

A geometria específica e o contraste da íris serão características biométricas utilizadas para medir o desempenho de leitura, entre eles a velocidade de leitura (número de palavras por minuto), o número de fixações (quantas vezes o aluno fixou o olhar durante a leitura do texto), o tempo médio de fixação e o número de regressões (quantas vezes o aluno voltou ao ler um texto).

Este trabalho tem como objetivo desenvolver um programa de rastreamento ocular, de baixo custo, para mensurar o desempenho de leitura. Poderá ser aplicado em escolas de ensino público, juntamente com a triagem da acuidade visual, para mensurar o desenvolvimento de leitura dos alunos no Brasil. Com esse novo método será possível registrar e armazenar em um banco de dados de maneira automática os dados de leitura dos alunos sem a intervenção do examinador. Durante a leitura será registrado a quantidade de fixação, o número de regressões, o tempo médio de fixação e a velocidade de leitura.

Para fazer o rastreamento aplicou-se a biblioteca OpenCv na implementação das operações básicas e fundamentais para o desenvolvimento de aplicativos na área da visão computacional, por possuir medidas de processamento de imagens e vídeos, em janelas independentes, com o comando do mouse ou teclado. Introduziu-se um software de reconhecimento de imagem em tempo real, através de um webcam ou câmera, sendo possível a retirada das informações contidas na imagem após o rastreamento da leitura.

O presente artigo encontra-se organizado da seguinte forma: revisão bibliográfica, em seguida os materiais e métodos e na última seção os resultados obtidos.

METODOLOGIA:

Para o desenvolvimento do aplicativo de rastreamento ocular foram utilizados os seguintes equipamentos:
• Câmera Jupiter, Modelo JP-638;
• Computador DELL Latitude E5420, processador Intel Core I5-2410M, 4GB Ram, HD 250 Gb;
• Placa de Captura de Vídeo PixelView PlayTV USB Hybrid.
E os seguintes aplicativos:
• Sistema Operacional Micrososft Windows 7 Professional;
• Intel OPENCV 2.3.1;
• Microsoft Visual Studio 2010 Express.

O software de processamento foi desenvolvido na linguagem C++ por permitir trabalhar em alto nível de forma rápida, também sendo possível empregar códigos que exijam desempenho em tempo real e por possuir integração com a biblioteca do Opencv.

A biblioteca Opencv foi escolhida por possuir diversas funções para cálculo matemático e processamento de imagens em aplicações de visão computacional em tempo real. Além disso, entre suas utilidades está à detecção de bordas, captura de objetos e reconhecimento de imagens, na qual, iremos trabalhar tendo como foco o desenvolvimento de aplicativos de visão computacional.

O vídeo utilizado nesse projeto foi captado com uma câmera de segurança do modelo JUPITER COLOR CCD, sendo capturado, primeiro, com a presença de luz (colorido). Também foi capturado com a ausência de luz visível, com iluminação infravermelho, para não influenciar a reação da pupila e destaca-la dos demais traços da face. O vídeo focaliza a face humana e tem duração de aproximadamente vinte e oito segundos e uma taxa de 29 quadros/s.
O primeiro vídeo foi convertido para o formato Lab e o segundo para o formato YCbCr com a intenção de destacar a pupila e obter pouca variação em sua detecção em ambos os casos.

Como a maioria das aplicações que utilizam visão computacional, inicialmente é necessário executar uma parametrização das características, como distância, cores, luz, etc. Deve-se estimar a direção do olho com a distância em que o usuário está olhando para uma determinada câmera fixa, para que ocorra o rastreamento ocular.

Para esta pesquisa, foi feito dois algoritmos para o processamento de Imagens aplicando-se até quatro filtros ou transformações: conversão do padrão de cores, separação e escolha de um canal de cor, suavização e detecção de bordas. Onde os algoritmos desenvolvidos são capazes de capturar imagens, reconhecer padrões identificando objetos previamente conhecidos e descritos, e rastrear os objetos, atualizando a cada frame o posicionamento da pupila de acordo com sua movimentação. Desta forma definem-se as coordenadas x e y do ponto que ajudarão para futuras análises referentes ao desempenho de leitura. Para Batista [3], para um pré-processamento ter sucesso é necessário uma simplificação da imagem ocasionando uma melhoria dos resultados.

Primeiro teste

Tem por finalidade comparar uma imagem principal e um modelo para combinar nas partes de um vídeo. O processo de MatchTemplate move a imagem do modelo para todas as posições possíveis de uma imagem original e calcula um índice numérico que indica a qualidade do modelo corresponde à imagem nessa posição. A detecção é feita em uma base de pixel por pixel. Em tempo real a imagem permanece em constante atualização, devido a isso, os valores do ponto de identificação da imagem resultante da comparação, se mantém sempre em atualização até o final do vídeo.
Neste primeiro teste, utilizou-se os seguintes comandos:

- cvMatchTemplate (const CvArr *imagem, const CvCrr *templ, CvCrr *result, int method): Usado para comparar a imagem de referência com o frame do vídeo.
- cvRectangle(IntPtr, MCvPoint, MCvPoint, MCvScalar, Int32, LINE_TYPE, Int32): Desenha um retângulo com dois cantos opostos.
- cvSetImageROI(IplImage *imagem, CvRect rect): Determina a região de interesse da imagem em um retângulo, diminuindo a região a ser pesquisada pelo comando MatchTemplate, resultando em um aumento de desempenho do aplicativo.

Segundo Teste

No segundo método utilizou-se os mesmos vídeos do primeiro teste, na qual foi utilizado um processo cíclico de captura e detecção, utilizando o comando cvHoughCircles. Obedecendo a uma frequência, uma imagem é capturada através de um vídeo (CvCaptureAVI e CvQueryFrame) e em caso de sucesso o vídeo colorido será tratado e convertido em três etapas: primeiramente em A (luminância), B e C (duas gamas de cor, cromoticidade). Já o vídeo com ausência de luminosidade será tratado e convertido em quatro etapas: primeiramente em gray (escala de cinza), segundo em Y (luminância) definindo a intensidade luminosa ou o brilho, terceiro em Cr (crominância azul) e por último em Cb (crominância vermelha). As Coordenadas da região onde os olhos foram detectados são comparadas ás regiões demarcadas no algoritmo.
Geralmente, necessita passar alguma forma de pré-processamento para redimensionar, realçar cores ou bordas ou ainda comprimir os dados.
As principais funções utilizadas com CvHoughCircles foram:

- cvCvtColor(IntPtr src, IntPtr dst,COLOR_CONVERSION code): Converte imagem de entrada a partir de um espaço de cor para outro.
- cvSplit(src, dst0, dst1, dst2, dst3): Divide um array multi-canal em matrizes de canal único separadas.
- cvSmooth (src, dst, tipo, param1, param2, param3, param4): Suaviza a imagem, reduzindo bordas indesejadas.
- cvSetImageROI(IplImage *imagem, CvRect rect): Determina a região de interesse da imagem em um retângulo.
- cvCanny (Imagem, bordas, threshold1, threshold2, aperture_size): Encontra as bordas da imagem, conforme apresentado na Figura 1.
- cvDilate (src, dst, element, iterations): Dilata a imagem.
- cvHoughCircles (IntPtr, IntPtr, HOUGH_TYPE, Double, Double, Double, Double, Int32, Int32): Encontra círculos em uma imagem em tons de cinza usando a transformada de Hough, conforme a Figura 2.

Figura 1: Frame após os comandos cvSetRoiImagem e cvCanny.

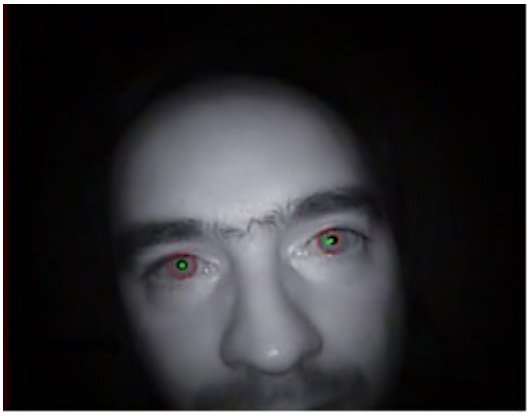

Figura 2: Olhos detectados após o comando cvHoughCircles.

RESULTADOS E DISCUSSÕES:

Utilizando o Match Template o processamento do vídeo ficou lento por ter que extrair as características da imagem de referência e assim fazer a marcação na imagem original. Com isso, o método utilizado vai para todas as posições possíveis de uma imagem original pegando vários traços da face, além dos olhos, devido ao fato do método buscar todos os índices numéricos que indica a qualidade do modelo correspondente à imagem nessa posição, dificultando a detecção da pupila.
No entanto tivemos que utilizar um método SetImageROI para determinar a região de interesse da imagem original, demarcando com um retângulo, assim diminuem os traços para fazer a comparação, deixando o vídeo mais rápido durante o processamento.
No gráfico, quando houver uma variação maior dos pontos, pode-se verificar uma falsa detecção da pupila. Analisando o gráfico 1, após o processamento do vídeo com presença de luz, observamos que dos 847 frames, 518 e 454 são pontos válidos na pupila direita e esquerda respectivamente. O restante dos pontos detectados é uma falsa detecção da pupila. Esse processamento ocorreu em um tempo de 488 segundos. Já o gráfico 2, após o processamento do vídeo com ausência de luz, observamos que dos 838 frames, 470 e 742 são pontos na região dos olhos esquerdo e direito respectivamente. Apesar da grande quantidade de pontos detectados na região dos olhos, durante a execução do processamento diversos pontos forma marcados fora da pupila / íris, podendo ter redução da confiabilidade dos pontos encontrados. Os pontos inválidos foram os detectados distantes da região ocular. Esse processamento ocorreu em um tempo de 491 segundos.
Comparando os gráficos 1 e 2, verificamos que o vídeo com presença de luz teve menor tempo, sendo assim maior velocidade de processamento. Entretanto obteve pior detecção das características e marcação da região de interesse, diminuindo as chances de sucesso de captação da pupila. Logo concluímos que mesmo o vídeo com ausência de luz obter o maior tempo de processamento, obteve melhor sucesso na detecção das características e captação da região de interesse.

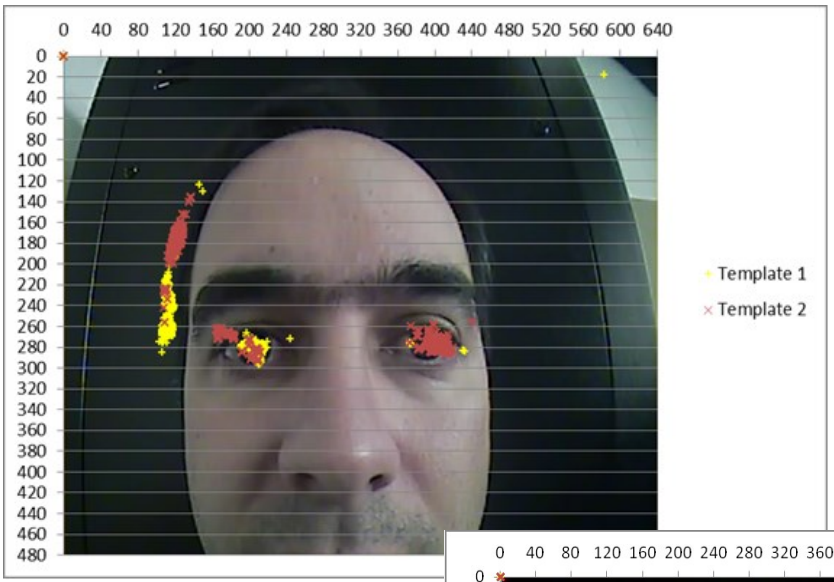

Gráfico 1: Detecção de pontos no vídeo com iluminação natural e técnica cvMatchTemplate.

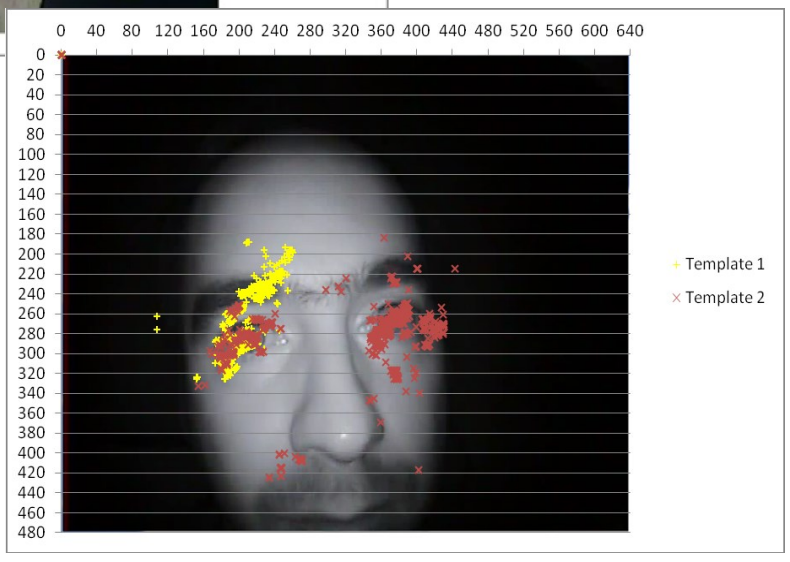

Gráfico 2: Detecção de pontos no vídeo com iluminação infravermelha e técnica cvMatchTemplate.

RESULTADOS E DISCUSSÕES:

No segundo teste, após a captura do vídeo, foi necessária uma conversão: o vídeo colorido foi convertido para Lab (figura 11) e o de ausência de luminosidade foi convertido para YCbCr (figura 12) com a intenção de diminuir contrastes permitindo um melhor rastreamento do objeto que deseja. Em seguida utilizou-se o comando HoughCircles que detectou os círculos nos frames, ou seja, a cada movimento dos olhos a pupila é detectada. Este comando melhorou a velocidade de processamento do vídeo.

Com a finalidade de destacar a borda da pupila utilizou-se o comando Canny para diminuir os traços da face, porém, alguns traços além da pupila ainda eram rastreados. Por isso foi necessário utilizar o método SetImageROI para determinar a região de interesse da imagem original, demarcando com um retângulo, diminuindo os traços desnecessários e consequentemente aumentando a velocidade dos vídeos.

Analisando o gráfico 3, após o processamento do vídeo colorido, observamos que dos 846 frames, 693 e 379 são pontos válidos na pupila direita e esquerda respectivamente. O restante dos pontos detectados é uma falsa detecção da pupila. Esse processamento ocorreu em um tempo de 100 segundos. Já o gráfico 4, após o processamento do vídeo com ausência de luz, observamos que dos 838 frames, 656 e 413 são pontos válidos na pupila esquerda e direita respectivamente. O restante dos pontos detectados é uma falsa detecção da pupila. Esse processamento ocorreu em um tempo de 100 segundos.

Comparando os gráficos 3 e 4, verificamos que ambos os vídeos tiveram o mesmo tempo de processamento. Entretanto o gráfico 3 obteve pior detecção das características e marcação da região de interesse, diminuindo as chances de sucesso de captação da pupila. Logo concluímos que o gráfico 4 obteve melhor sucesso na detecção das características e captação da região de interesse pois seus pontos ficaram mais concentrados.

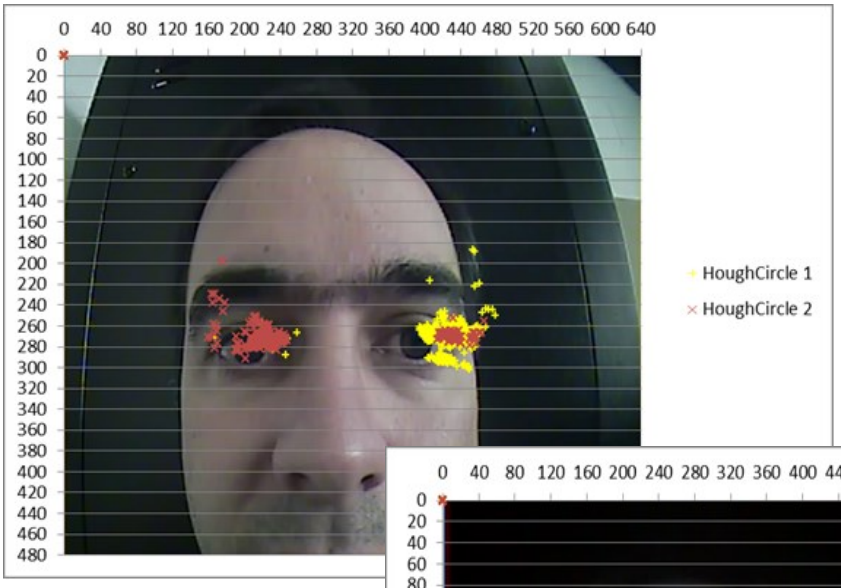

Gráfico 3: Detecção de pontos no vídeo com iluminação natural e técnica cvHoughCircle.

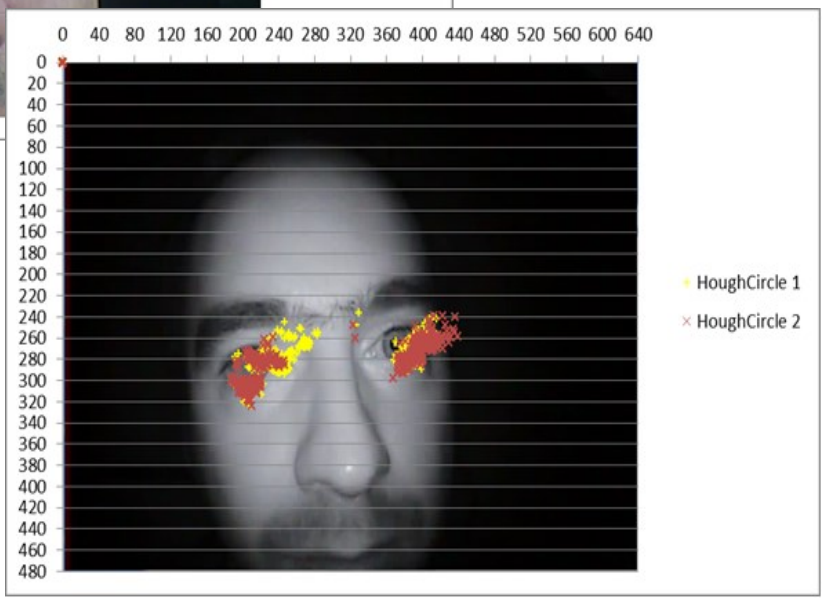

Gráfico 4: Detecção de pontos no vídeo com iluminação infravermelha e técnica cvHoughCircle.

RESULTADOS E DISCUSSÕES:

Ao comparar os resultados, podemos observar que para a técnica de detecção de círculos a quantidade de dados encontrados foi semelhante para os vídeos colorido e tons de cinza. Para a técnica de detecção de padrão, apesar de ter detectado um maio número de pontos no vídeo em tons de cinza, a marcação foi menos precisa, ou seja, está mais dispersa do que a utilizando a técnica de detecção de círculos.

Tabela 1: Dados do Processamento do Primeiro e Segundo Teste.

VÍDEO	Nº DE FRAMES	Nº PONTOS DETECTADOS			TEMPO (s)
		OLHO DIREITO	OLHO ESQUERDO	TOTAL	
Match Template Colorido	847	518	454	972	488
Match Template Tons de Cinza	838	470	742	1212	491
HoughCircle Colorido	846	693	379	1072	100
HoughCircle Tons de Cinza	838	656	413	1069	100

CONCLUSÕES:

No desenvolvimento de um sistema de rastreamento ocular devemos adquirir um algoritmo com funções que tenham maiores velocidades no processamento e pouca variação nas coordenadas espaciais (x, y), pois são estes que irão mensurar o desempenho da leitura.

Com esse trabalho foi possível identificar que o melhor programa para detectar a pupila do olho com menor dispersão e maior confiabilidade foi utilizando a detecção de círculos com o comando CvHoughCircles. Já os pontos relacionados com o algoritmo MatchTemplate, ocorreu maior dispersão dos pontos, e, além disso, seu tempo de processamento foi superior. Para trabalhos futuros deve ser utilizada uma câmera com melhor resolução em relação da que foi utilizada, melhorando a detecção em ambas as técnicas e o desenvolvimento do sistema relacionando a leitura de um texto com a posição dos olhos.

REFERÊNCIA BIBLIOGRÁFICA:

[1] LORIGO, L., Haridasan, M., Brynjarsdóttir, H., Xia, L., Granka, L., Pellacini, F., Pan, B., Joachims, T., Gay, G. (2008). " Eye-Tracking and online search: lessons learned and challenges ahead ". Journal of the American Society for Information Science and Technology.

[2] MANZI A. F. "Aplicação de Visão Computacional para Extração de Características em Imagens do Olho Humano“ São Paulo: Trabalho de Conclusão de Curso de Engenharia Mecânica apresentado á USP, 2007.

[3] MINISTÉRIO DA SAÚDE. Secretaria de Atenção à Saúde. " Projeto Olhar Brasil ". Brasília, 2007. Relatório, 24p.

[22] SOARES, C.F. Sistema de Triagem Visual e Auditiva de crianças em idade escolar, conectado a um banco de dados. Belo Horizonte: Tese de Doutorado em Engenharia Mecânica apresentado á UFMG, 2009.

Participação em Congressos, publicações e/ou pedidos de proteção intelectual:

Este trabalho foi apresentado no ano de 2013 em:

- XI Congresso Ibero-Americano de Engenharia Mecânica
- Semana de Iniciação Científica do Campus Congonhas.