

Desenvolvimento de um esquema simplificado de recepção do *Optical Camera Communications (OCC)* usando dispositivos Android

Palavras-Chave: *Optical Camera Communications, Screen-to-Camera, Visible Light Communications (VLC)*

Autores:

Matheus Fortunato Alves [UNICAMP/FT]

Prof. Dr. Leandro Ronchini Ximenes (orientador) [UNICAMP/FT]

INTRODUÇÃO

O surgimento de novas tecnologias como o 5G e a Internet das Coisas (IoT) prevê um aumento massivo no fluxo de dados gerado nos diversos sistemas de comunicações. Até então, o espectro de Radiofrequência (RF - *Radio Frequency*), em especial a faixa entre 300 kHz e 30 GHz, tem sido usado para implementar as redes de telecomunicações móveis sem fio em larga escala do planeta. Entretanto, devido à escassez do espectro licenciado, ao alto custo na produção de *hardware* na faixa de RF, além da interferência entre diferentes tecnologias sem fio (e.g., WiFi, Bluetooth, entre outros), surgiu-se a necessidade de tecnologias alternativas que pudessem solucionar alguns destes desafios impostos pela nova geração da comunicação móvel [1].

Neste contexto, a Comunicação Óptica sem Fio (OWC - *Optical Wireless Communication*) se mostrou promissora, utilizando a faixa de espectro óptico para transmitir dados. Esse espectro é dividido em 3 grandes faixas: luz visível (VL - *Visible Light*), infravermelho (IR - *Infra Red*) e ultravioleta (UL - *Ultra Violet*). Todo enlace OWC consiste em uma fonte óptica transmissora de dados, o meio livre como canal de comunicação e um detector óptico como receptor [2]. A tecnologia que usa a faixa de VL, entre 400 THz e 750 THz, é denominada Comunicação por Luz Visível [3] (VLC - *Visible Light Communication*) e emprega normalmente Diodos Emissores de Luz (LEDs - *Light Emitting Diodes*) como fontes transmissoras e Fotodetectores (PD - *Photodetectors*) como receptores. Um dos aspectos mais atrativos da comunicação VLC está na dualidade de funções,

já que um transmissor VLC pode ser usado para as tarefas simultâneas de comunicação e iluminação.

Inserido no escopo do VLC, a tecnologia que utiliza um Sensor de Imagem (IS - *Image Sensor*) de uma câmera como fonte receptora ao invés de um fotodiodo é denominada Comunicação por Câmera Óptica (OCC - *Optical Camera Communications*) [4]. A Figura 1 ilustra um diagrama de blocos do sistema OCC do tipo *Screen-to-Camera* (S2C), que emprega telas e displays LED como transmissores ao invés de LEDs discretos.

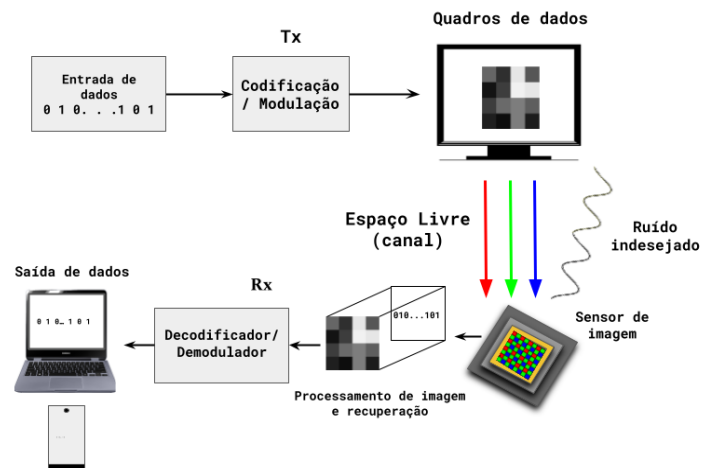


Figura 1: Diagrama de blocos de um típico sistema de comunicação OCC.

Pode-se dizer que os principais aspectos que trouxeram notoriedade ao OCC se sustentam na popularização dos *smartphones* com câmera embutida e outras câmeras comerciais de baixo custo [5]. Aliado a este fato, a infraestrutura de iluminação preexistente baseada em LEDs permite que a implementação desse tipo de tecnologia tenha um baixo custo de implementação, pois

tanto o transmissor quanto o receptor já estão disponíveis em larga escala.

Dentre os principais aspectos técnicos positivos do OCC, destacam-se uma alta Relação Sinal-Ruído (SNR - *Signal-to-Noise Ratio*), robustez às variações da distância de comunicação, alta mobilidade, segurança aprimorada e reduzida interferência em relação a outros sistemas de comunicação [1]. Mesmo com todos esses aspectos a favor, o OCC possui algumas desvantagens, como a baixa taxa de dados, limitada pela baixa taxa de quadros da maioria das câmeras comerciais. Outro desafio da tecnologia OCC reside nos problemas de sincronia entre a taxa de símbolo enviada pelo transmissor e a taxa de captura da câmera.

Este projeto de Iniciação Científica (IC) desenvolveu e implementou uma plataforma de recepção OCC flexível do tipo S2C. Esta plataforma permite a implementação prática de diversos sistemas de comunicação OCC, facilitando a validação experimental de resultados de pesquisa na área.

O desenvolvimento da plataforma foi dividido em duas etapas. Primeiro foi desenvolvido um aplicativo Android com uma Interface Gráfica de Usuário (GUI - *Graphical User Interface*), que permite alterar os principais parâmetros da câmera que influenciam o desempenho de recepção do sistema OCC proposto e realizar a aquisição de vídeos para posterior processamento. Então experimentos com vídeos reais e uma *smartphone* foram usados para validar a funcionalidade do esquema proposto. A câmera captura o vídeo transmitido pela tela, que possui dados (*bits*) codificados em cada um dos seus pixels. Depois, com o auxílio de técnicas de processamento de imagens e de etapas de demodulação de sinais, esses dados são recuperados quadro a quadro. O desempenho do sistema proposto foi verificado através de curvas de erro de estimação dos vídeos e dados transmitidos.

As seções seguintes ilustram os métodos utilizados na construção do aplicativo da câmera e dos algoritmos desenvolvidos. Também são exibidas informações referentes à rotina de testes e resultados do esquema de recepção do sistema OCC.

METODOLOGIA

O aplicativo que faz a interface gráfica do receptor foi desenvolvido com o auxílio da plataforma *Android Studio*, na linguagem de programação *Java*. Após a criação do aplicativo, desenvolveu-se rotinas de processamento de imagem no *software* MATLAB para tratamento dos quadros adquiridos pela câmera, assim como para a subsequente detecção dos dados transmitidos.

A geração dos dados para transmissão, bem como os algoritmos de modulação/demodulação e estimação dos parâmetros foram desenvolvidos em projeto de pesquisa do aluno de mestrado Bruno de Almeida Larêdo, sob orientação do Prof. Dr. Leandro Ronchini Ximenes, da Faculdade de Tecnologia (FT).

Na primeira parte do projeto foi criado o aplicativo que grava vídeos, armazena-os num cartão de memória e permite a modificação de alguns parâmetros da câmera como: taxa de quadros do vídeo de saída (*frame rate*), resolução do vídeo, taxa de captura (*capture rate*) e o foco da câmera. Além disso, ele permite a opção de ativação/desativação de alguns recursos, como estabilização de vídeo, exposição automática e nível de balanço de branco. Para desenvolver o aplicativo, a API *Camera* foi a principal biblioteca usada para manipular a câmera e seus parâmetros, assim como pelo acesso e controle da câmera do dispositivo.

No aplicativo, inicialmente, a câmera é reconhecida e declarada. Em seguida, um plano retangular é criado para suportar a visualização da câmera na tela do dispositivo. Depois, desenvolveu-se uma forma de manipulação dos parâmetros da câmera. Também, um botão que permite iniciar e interromper a gravação foi inserido. Após o fim da gravação, o vídeo é salvo no formato *.mp4* e nomeado de acordo com uma formatação característica dos arquivos *Android*.

Na segunda parte do projeto foi desenvolvido um algoritmo que processa o vídeo capturado pelo aplicativo da câmera e tenta reconstruí-lo para que se torne o mais próximo possível do vídeo original, possui dimensões espaciais J por L e F quadros (*frames*). Para os testes de validação, cada pixel do vídeo foi

modulado com uma mensagem (sequência de *bits*) aleatória.

Ao passar pelo canal (i.e., propagação no espaço livre), nem sempre o IS consegue recuperar com precisão a informação transmitida. Diversos fatores de degradação do vídeo estão presentes, como borramento (*blurring*), distorção de perspectiva, interferência por ruído térmico ou luz ambiente, assim como os problemas de sincronia entre o transmissor e o receptor. Por isso, é necessário usar técnicas de recuperação e reconstrução de imagem para obter um desempenho satisfatório na recepção dos dados.

Perceba também que as cenas gravadas pelo aplicativo possuem vários elementos indesejados, que não são usados no processo de detecção de bits, como o teclado do *notebook* e os outros elementos da tela em si. Por isso, foi utilizada uma técnica manual de detecção de Região de Interesse (RoI - *Region of Interest*).

Mesmo após o recorte da RoI, devido à resolução da câmera utilizada e da distância entre transmissor e receptor, entre outros fatores, os quadros do vídeo capturado ainda possuíam dimensões espaciais distintas do vídeo original, isto é, $J \times L$. O redimensionamento do vídeo capturado para as dimensões do vídeo original foi feito através da segmentação dos pixels da RoI em $J \times L$ blocos. Para cada bloco, a intensidade do pixel do vídeo redimensionado foi feita pela média aritmética não-ponderada dos pixels daquele bloco.

Devido a erros de sincronia, e às diferenças entre os intervalos de transmissão e recepção, notou-se também que o vídeo capturado possuía mais quadros que o original. Por isso, optou-se por adotar uma técnica de sinalização para identificar o primeiro quadro do vídeo transmitido. Como os primeiros quadros do vídeo transmitido eram todos brancos (i.e., todos os pixels com intensidade máxima), foi utilizada a comparação das variâncias dos pixels entre os quadros. Assim, ao notar uma diferença de variância alta, o receptor consegue perceber o início efetivo da transmissão dentro dos quadros do vídeo capturado. De forma semelhante, uma vez que o vídeo capturado se encerra instantes após o fim do vídeo transmitido, o último quadro deste também foi encontrado pela mudança brusca da variância.

Em seguida, um processo de normalização das intensidades de *pixel* de cada quadro desse vídeo, isto é, as intensidades de pixel de cada quadro foram ajustadas dentro do intervalo possível de valores na escala de cinza, i.e., branco (intensidade máxima) e preto (intensidade mínima)

Mesmo com o ajuste de sincronia para detectar o início e o fim da transmissão, verificou-se que o número quadros do vídeo não estava de acordo com o vídeo original, devido aos descompassos entre a taxa de reprodução do vídeo e a taxa de captura da câmera. Este problema de sincronia em sistemas OCC é bem conhecido [4], já que tanto a taxa de reprodução pode ser afetada pelo tempo de processamento no transmissor, quanto a taxa de quadros na saída da câmera pode ser afetada tanto por questões de hardware ou de *software* da mesma.

Para os testes realizados, verificou-se que, em geral, a taxa de captura era ligeiramente maior que a taxa de reprodução, o que gerava quadros extras indesejados no vídeo reconstruído. Por isso, foi aplicada uma técnica adicional de detecção e correção de sincronia baseada na correlação entre os quadros do vídeo original e do vídeo reconstruído. Graças à alta SNR dos testes, típica em sistemas de comunicação OCC, caso o quadro do vídeo recuperado apresentasse um fator de correlação de Pearson menor que 0,8 em relação ao quadro do vídeo original, ele seria descartado. Após essa correção, um novo vídeo era gerado, exatamente com o mesmo número de quadros do vídeo original.

A Figura 2 exhibe a modificação gerada em um dos quadros do vídeo capturado a cada etapa do algoritmo que foi descrita, até a reconstrução e comparação com esse mesmo quadro do vídeo original. Para este exemplo, foi utilizado um vídeo de baixíssima resolução, de 4 por 4 *pixels*. Esta figura é um resultado obtido experimentalmente e serve apenas para ilustrar a metodologia desenvolvida. Note que o quadro do vídeo reconstruído (Fig. 2.(e)) não é exatamente igual ao quadro do vídeo original, (Fig. 2.(a)), já que existe a presença de ruído e outras interferências.

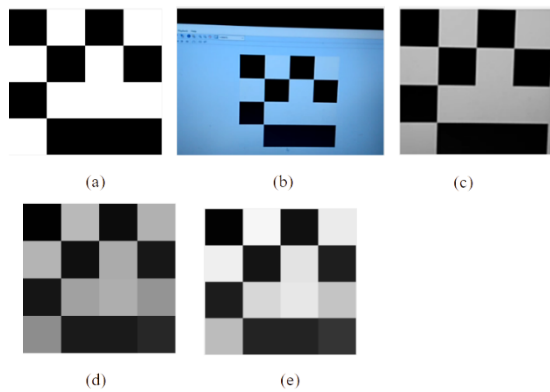


Figura 2: (a) - Quadro original; (b) - Quadro capturado pelo aplicativo da câmera; (c) – Identificação manual da RoI em (b); (d) - Quadro redimensionado; (e) - Quadro normalizado.

Por último, usando o vídeo reconstruído em rotina pronta de detecção de sinais, foi possível obter a sequência de *bits* codificada nos pixels do vídeo transmitido. Foi possível obter então as curvas de erro quadrático médio normalizado (*NMSE – Normalized Mean Square Error*) dos símbolos detectados e do vídeo reconstruído, usadas para validar a funcionalidade da plataforma desenvolvida. Alguns destes resultados são apresentados na seção a seguir.

RESULTADOS E DISCUSSÕES

Para a validação da plataforma desenvolvida através de testes experimentais, foram gerados vídeos codificados com símbolos aleatórios. Para a implementação de testes usando o sistema completo OCC foi utilizado o display de um *notebook* Dell Inspiron 5458 como transmissor e um *smartphone* Moto G7 play da Lenovo como receptor. A Figura 3 mostra o aplicativo aberto no *smartphone*, no *setup* experimental de validação da plataforma. A modulação utilizada foi a Modulação por Posição de Pulso (PPM – *Pulse Position Modulation*) [3], com $J = L = 4$, que confere ao quadro da Figura 3 o aspecto de um “tabuleiro de xadrez”.

As curvas usadas para validação foram geradas em função de dois parâmetros distintos que controlam o SNR de recepção.

O parâmetro de SNR era ajustado artificialmente na própria codificação dos vídeos, com um ruído branco e gaussiano sendo adicionado por *software*.

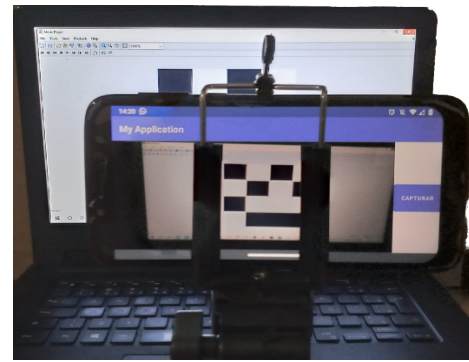


Figura 3: Visualização do *app* e esquema proposto.

A Tabela 1 exibe os parâmetros utilizados nos transmissor e receptor para teste do sistema.

Tabela 1: Parâmetros do transmissor e receptor

Receptor: Moto G7 Play	Transmissor: Dell Inspiron 5458
<i>Frame rate</i> : 15 fps	<i>Frame rate</i> : 15 fps
Resolução : 320 x 240 (padrão QVGA)	Resolução : 4 x 4
Número de quadros : Variável	Número de quadros : 160

Foram usados quatro níveis diferentes de SNR no teste: 0, 20, 40, 60 e 80 dB. Para cada valor de SNR testado, um conjunto de capturas de vídeo foram realizadas alterando-se o parâmetro de brilho tela do *notebook* em 100%, 50% e 25% do brilho total. Todos os testes foram feitos em condições similares de luminosidade, em um quarto residencial escuro (e.g., luzes artificiais apagadas e cortinas fechadas).

A Figura 5 e a Figura 6 apresentam os níveis de NMSE de vídeo e de símbolo, respectivamente, calculados com e sem o ajuste de sincronia de quadros descrito na seção de Metodologia. Fica evidente pela Figura 5 que, sem o esquema de ajuste proposto, a recuperação e restauração do vídeo original ficam fortemente prejudicados, uma vez que a curva vermelha cai apenas lentamente com o SNR (dB). A mesma conclusão é válida para o erro de símbolo (Figura 6), que indica um efeito benéfico na estimação de bits quando o método de ajustes de sincronia é utilizado. Note também que há uma redução considerável de ambos os erros com o SNR, que era previsto na teoria, o que valida a

funcionalidade da plataforma proposta, embora ainda haja distorções e não-idealidades que não foram corrigidas.

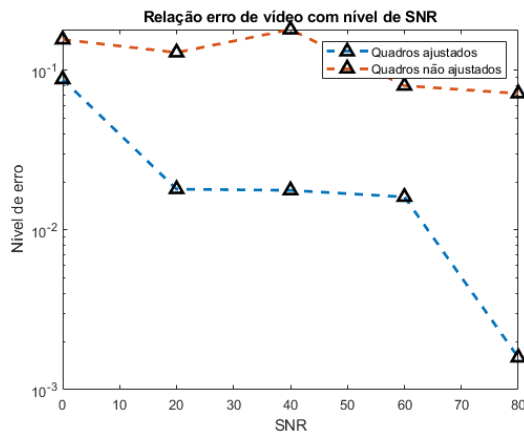


Figura 5: NMSE de vídeo, com e sem ajuste de quadros

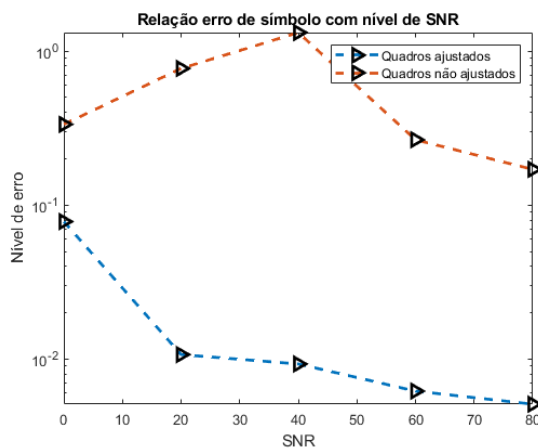


Figura 6: NMSE de símbolo, com e sem ajuste de quadros

Na Figura 7 o comportamento do erro da matriz de símbolos transmitida para diferentes níveis de SNR aliadas ao brilho da tela transmissora é mostrado. Dele, é possível notar que, conforme o brilho diminui, os níveis de erro de símbolo aumentam visto que a intensidade luminosa disponível no receptor OCC é reduzida. Uma vez que o brilho do LED transmissor é proporcional à potência do sinal, e, portanto, ao SNR, aumentar o brilho tem efeito análogo ao aumento artificial do SNR no vídeo codificado. Para se atingir um nível de erro de 10^{-2} , dobrar o brilho da tela do notebook teve efeito similar a usar um SNR 30 dB maior. Em outras palavras, o desempenho do sistema a 100% do brilho foi similar ao desempenho com 50% do brilho se a potência do ruído fosse 30dB menor.

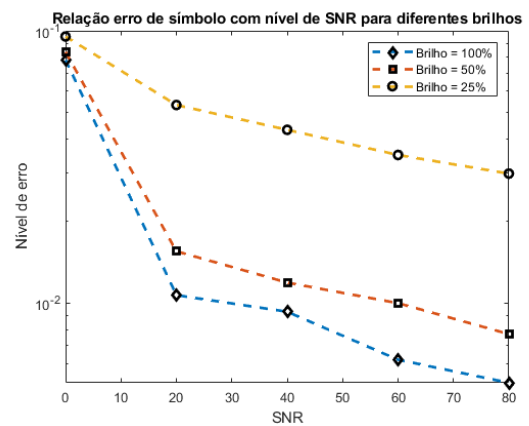


Figura 7: Comparação entre erros de símbolo para diferentes níveis de brilho.

CONCLUSÃO E PERSPECTIVAS

Este projeto desenvolveu e implementou uma plataforma funcional de recepção em sistemas OCC a partir do desenvolvimento de um aplicativo *Android* e de códigos de processamento de imagens no receptor.

A plataforma é compatível com diversos esquemas de modulação e codificação para sistemas OCC. Foram feitos testes de validação da plataforma, através da montagem de um setup composto por um *notebook* (transmissor) e um *smartphone* (receptor). Os resultados mostram que o sistema é funcional, com desempenho esperado pelo projeto.

BIBLIOGRAFIA

- [1]: M. Z. Chowdhury, M. T. Hossan, A. Islam and Y. M. Jang, "A Comparative Survey of Optical Wireless Technologies: Architectures and Applications," in *IEEE Access*, vol. 6, pp. 9819-9840, 2018.
- [2]: Z. Ghassemlooy, S. Arnon, M. Uysal, Z. Xu, and J. Cheng, "Emerging optical wireless communications-advances and challenges," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 9, pp. 1738-1749, Sep. 2015.
- [3]: P. H. Pathak, X. Feng, P. Hu and P. Mohapatra, "Visible Light Communication, Networking, and Sensing: A Survey, Potential and Challenges," in *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2047-2077, Fourth Quarter 2015, doi: 10.1109/COMST.2015.2476474.
- [4]: T. Nguyen, A. Islam, T. Hossan and Y. M. Jang, "Current Status and Performance Analysis of Optical Camera Communication Technologies for 5G Networks," in *IEEE Access*, vol. 5, pp. 4574-4594, 2017, doi: 10.1109/ACCESS.2017.2681110.
- [5]: S. Arnon, Visible Light Communication. Cambridge, U.K.: Cambridge Univ. Press, 2015.