

Lista de Exercícios de Sistemas de TV – 2019-1

Luiz Felipe da S. Coelho
lfscoelho@ieee.org
DETEL – UERJ

Departamento de Engenharia Eletrônica e Telecomunicações
Universidade do Estado do Rio de Janeiro

14 de Maio de 2019

1 Informação de Cor

Objetivo: O principal objetivo das atividades desta seção é verificar empiricamente os aspectos dos sistema psico-visual humano no que se refere à percepção de cor. Temos ainda como objetivos a fixação do conceito e o uso de histogramas bem como o emprego de filtros bidimensionais. O secundário é expandir os conhecimentos sobre manipulação de matrizes e exibição de imagens usando o Matlab.

1.1 Espaços de Cores

1. **Tarefa:** Para cada uma das imagens CAPE, MANDRILL, CLOWN e TREES (que vão anexas), fazer o seguinte:
 - (a) Carregá-la e visualizá-la,
 - (b) Transformá-la de RGB para YIQ e YCbCr,
 - (c) Mostre as matrizes de R, G e B juntamente com as de Y, I e Q e as de Y, Cr e Cb, como se cada uma fosse uma imagem separada em níveis de cinza.

Dica: Há diferentes versões de fórmulas de conversão de RGB para YCbCr e de volta, veja que as fórmulas das recomendações ITU-R BT.601 (vídeo SD) e ITU-R BT.709 (vídeo HD), use os propostos na última.

Dica: Use o comando `subplot` para mostrar as imagens numa mesma figura.

Dica: Note que os elementos de I e Q podem assumir valores negativos – assim, faça a “renormalização necessária”, veja o comando `imgecsc`.

2. **Tarefa:** Apresente uma função que realize essas conversões (RGB↔YIQ, RGB↔YCbCr, YCbCR↔YIQ) recebendo como parâmetro a imagem alvo armazenada como dado volumétrico (isto é, uma matriz $L \times C \times 3$, onde L é a quantidade de linhas da imagem, C é a quantidade de colunas e 3 se refere às três componentes) o espaço de cores da imagem de entrada e o espaço de cores da imagem de saída; e cuja saída seja a imagem no espaço de cores desejado.

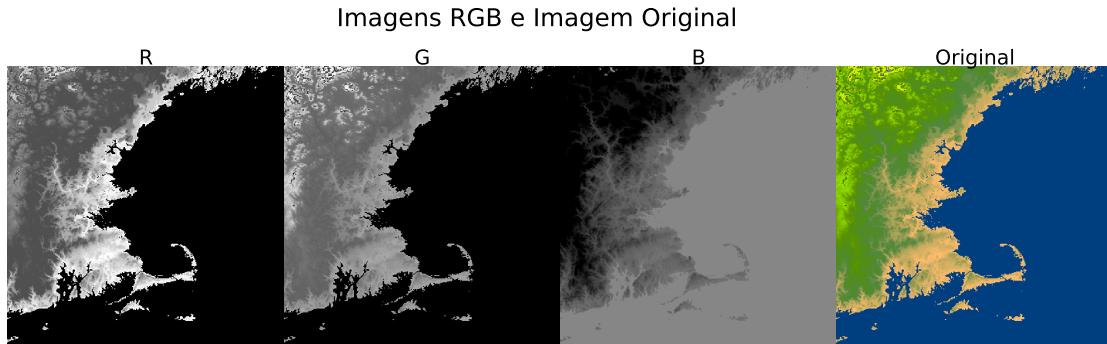


Figura 1: Imagem cape.tif, dividida em níveis RGB, juntamente com a imagem original.

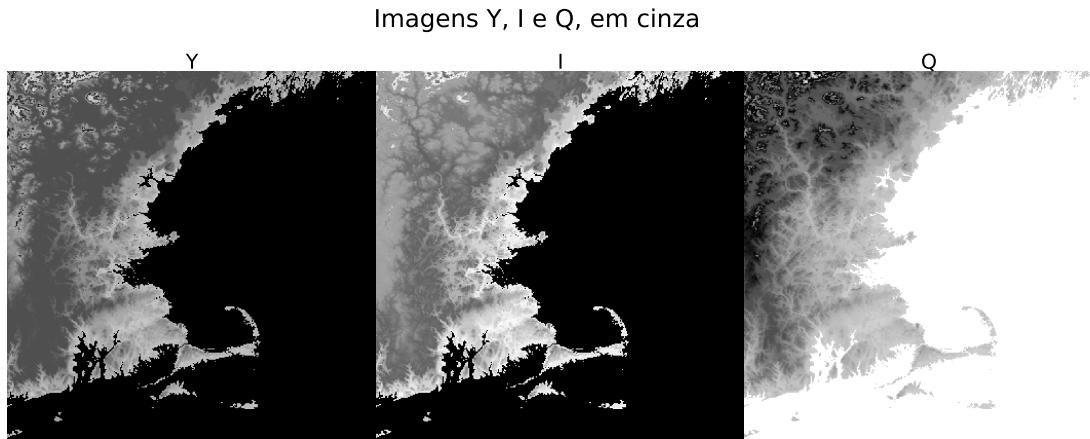


Figura 2: Imagem cape.tif, dividida em níveis YIQ.

Dica: Você deve primeiro fazer esta função e depois realizar a sequência de passos acima e as análises relevantes.

As funções pedidas estão no *script imgconvfunc.py*.

1.2 Resposta do Sistema Visual à Informação de Cor

1. **Tarefa:** Faça uma função que receba uma imagem I e retorne o histograma da luminância da mesma. O histograma de uma variável aleatória é a contagem das aparições / realizações de cada um dos valores que a variável pode assumir. Em nosso caso, isso pode ser representado por dois vetores X e Q , onde X indica os valores que as luminâncias dos pixels de I podem assumir e Q é a frequência ou quantidade de ocorrências de cada um dos valores de X em I .

Obs: Assuma que a imagem só possui valores inteiros.

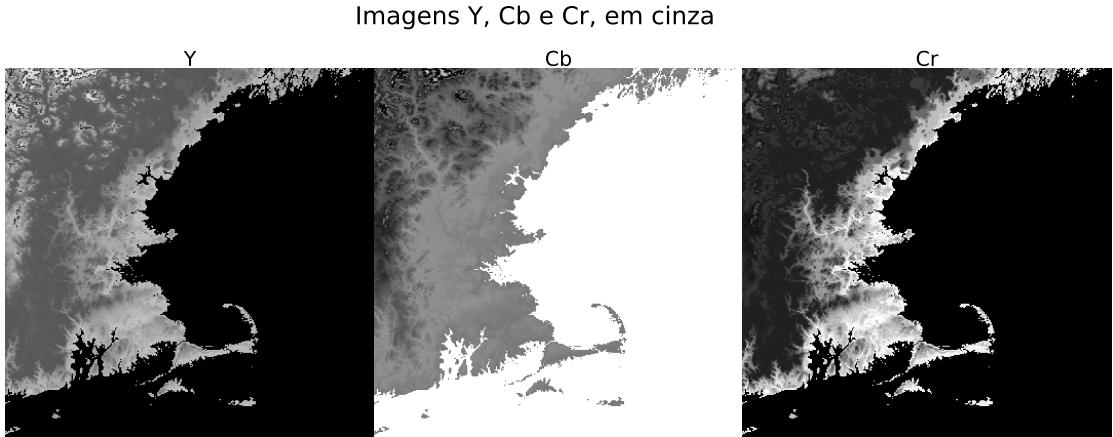


Figura 3: Imagem cape.tif, dividida em níveis YCbCr.

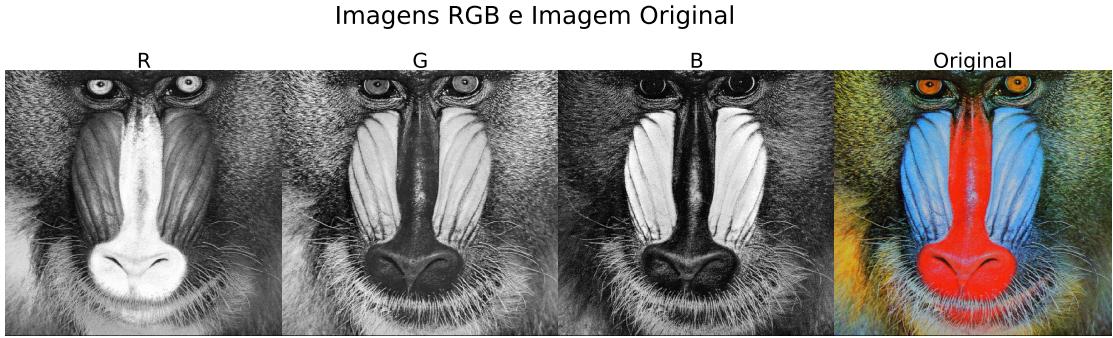


Figura 4: Imagem mandrill.tif, dividida em níveis RGB, juntamente com a imagem original.

Dica: Seja X uma variável aleatória discreta que pode assumir os valores $\{x_0, x_1, x_2, \dots, x_{N-1}\}$, o histograma para um conjunto de realizações ou amostras K dessa variável aleatória é uma tabela que indica quantas ocorrências de cada um dos N valores que X pode assumir há nas K ocorrências. Isto é, o histograma fornece a frequência relativa dos possíveis valores da variável aleatória X .

Seja X um vetor de comprimento K , teremos

$$\text{histograma}_X(x_i) = \#\{X == x_i\}, \quad (1)$$

onde $\#\{\cdot\}$ indica a quantidade, e $==$ é um operador lógico que indica igualdade.

Dica: Para encontrar as ocorrências de um dado valor x_i em uma imagem I no MatLab podemos usar o comando `find`, seguido da contagem dos elementos encontrados usando a função `size`.

Imagens Y, I e Q, em cinza

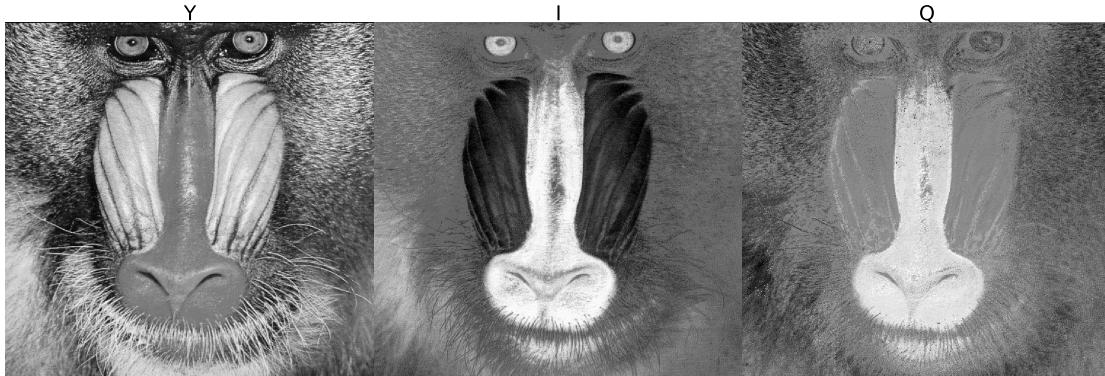


Figura 5: Imagem mandrill.tif, dividida em níveis YIQ.

Imagens Y, Cb e Cr, em cinza

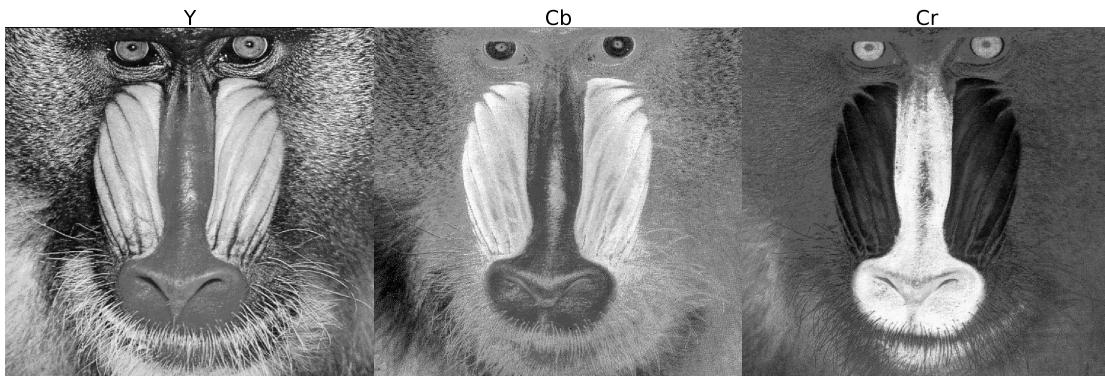


Figura 6: Imagem mandrill.tif, dividida em níveis YCbCr.

A função que realiza o histograma desejado está no *script funcs32.py*.

2. **Pergunta:** O Matlab fornece o comando `hist` para computar um histograma. Qualquer dúvida use o “help” do Matlab, por exemplo, via `help hist`. Compare as diferenças entre essa função e a que você implementou no item acima.

A diferença principal entre os dois algoritmos é que o histograma realizado por no item acima assume que a imagem possui apenas valores inteiros e realiza o histograma para todos os valores possíveis. Enquanto que o comando `hist` do Matlab realiza o histograma para apenas 10 níveis.

3. **Tarefa:** Podemos realizar histogramas de qualquer variável. Assim, compute os histogramas das componentes dos espaços de cores RGB, YIQ, e YCbCr (considerando

Imagens RGB e Imagem Original



Figura 7: Imagem clown.tif, dividida em níveis RGB, juntamente com a imagem original.

Imagens Y, I e Q, em cinza

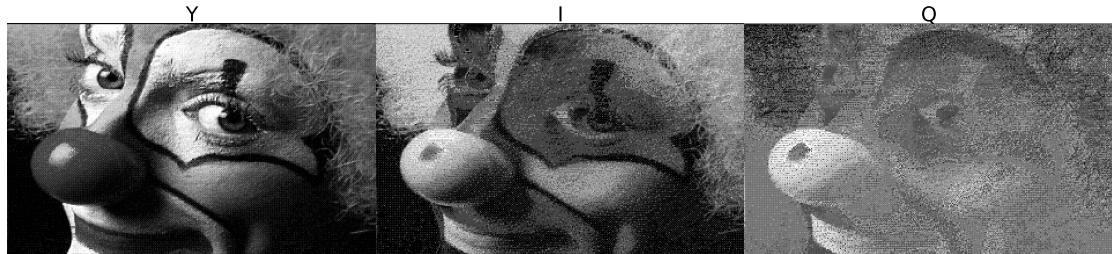


Figura 8: Imagem clown.tif, dividida em níveis YIQ.

256 níveis) para cada uma das imagens aqui consideradas e a partir desses histogramas calcule a entropia das diferentes componentes nos diferentes espaços de cores.

O *script funcs32.py* contém a função que realiza o cálculo de entropia, baseado no histograma calculado.

4. **Pergunta:** O que você observa a partir dos histogramas acima? Explique.

As imagens têm predominantemente termos em uma mesma faixa do histograma, os termos são, em sua maioria, distribuídos em um ou dois valores.

5. **Pergunta:** Comparando a entropia das diferentes imagens nos diferentes espaços de cores, o que você conclui? Discuta as vantagens e desvantagens de cada representação / espaço de cores em função da origem / tipo da imagens e suas necessidades de transmissão e armazenamento.

Os espaços de cores possuem diferentes valores de entropia. As informações RGB estão mais distribuídas entre os níveis de cores, ou seja, a entropia muda de acordo com as

Imagens Y, Cb e Cr, em cinza

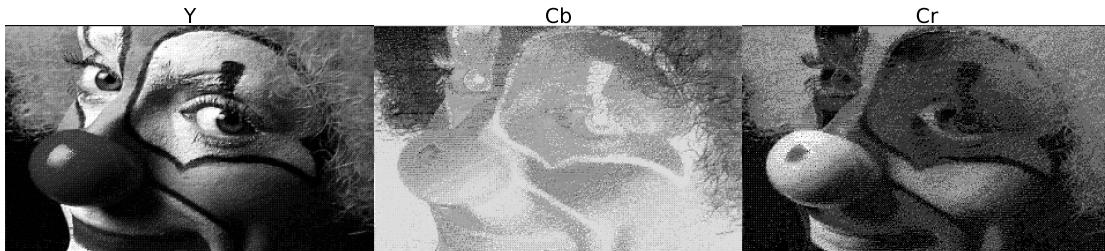


Figura 9: Imagem clown.tif, dividida em níveis YCbCr.

Imagens RGB e Imagem Original



Figura 10: Imagem trees.tif, dividida em níveis RGB, juntamente com a imagem original.

cores da imagem (por exemplo `clown.tif` possui mais entropia em R, enquanto que `cape.tif` possui mais informações em B). Já os espaços YIQ e YCbCr possuem maior entropia no espaço Y.

A entropia indica a quantidade de informação que a imagem carrega, portanto em uma necessidade de transmissão do caso RGB os espaços de cores têm importância equivalente, enquanto que os casos YIQ e YCbCr apresentam maior importância no espaço Y.

1.3 Visualização de Espaços de Cores

1. **Tarefa:** Gere três matrizes R, G e B (de 512 colunas por 100 linhas) com valores variando linearmente entre 0 e 255 com passo 1.
2. **Tarefa:** Utilize-as para tentar visualizar a faixa de cores que o monitor é capaz de representar de cada uma das cores R, G e B, isto é separadamente.

Imagens Y, I e Q, em cinza

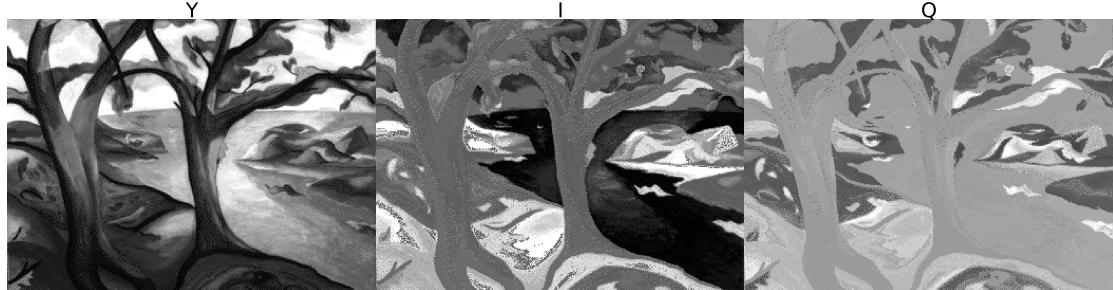


Figura 11: Imagem trees.tif, dividida em níveis YIQ.

Imagens Y, Cb e Cr, em cinza

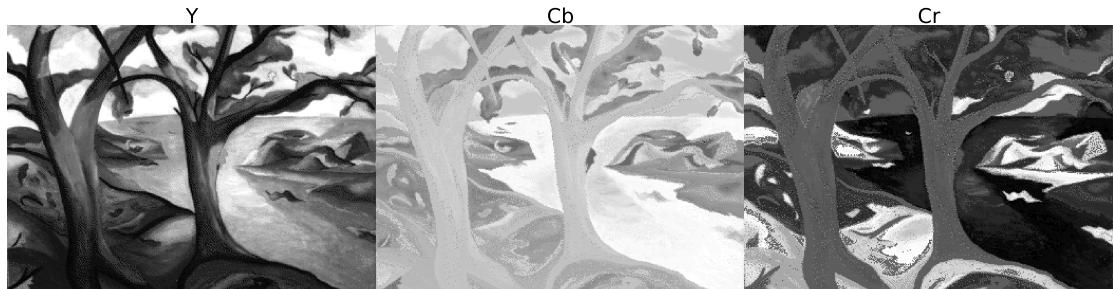


Figura 12: Imagem trees.tif, dividida em níveis YCbCr.



Figura 13: Níveis R, G e B divididos com as cores para 8-bits

3. **Pergunta:** Explique de que forma você realizou a tarefa acima!

Para realizar esta tarefa, dividi os espaços de cores em imagens R, G e B de maneira que o arranjo R possua valores não nulos apenas na matriz equivalente aos tons de

vermelho, fiz o equivalente para os arranjos G e B.

4. **Pergunta:** Se ajustarmos algum parâmetro do monitor, a faixa de cores visualizável pode se alterar? Qual parâmetro? Como se dá essa mudança? Explique e Justifique!

Sim. Se alterarmos o parâmetro matriz do monitor alteramos a faixa de cores que o monitor é capaz de representar.

5. **Tarefa:** Faça o mesmo nos espaços YIQ e YCbCr. Considerando $Y=128$ (isto é um nível de luminância médio) apresente as faixas de cores que as componentes podem assumir nos diferentes espaços de I e Q, e Cb e Cr.

Dica: Considere o plano IQ (ou CbCr), para cada par ordenado (I, Q) podemos encontrar a cor definida por $(Y=128, I, Q)$ (ou $(Y=128, Cb, Cr)$) em RGB. Podemos fazer isso para todos os pontos do plano. Assim teremos para cada pixel o valor RGB da cor. Podemos assim plotar o plano com as cores definidas pelas coordenadas usadas nas ordenadas e abcissas. Repare que isto é um corte no nível $Y=128$ no cubo de cor.

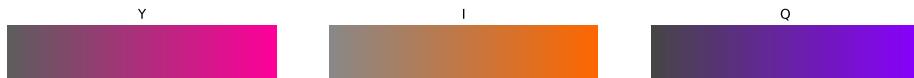


Figura 14: Cortes em $Y=128$, para I variando nos 8-bits, Q também variando nos 8-bits e Y representa I e Q variando nos 8-bits juntos. As matrizes foram convertidas para o formato RGB para poderem representar as cores que cada nível é capaz de produzir.

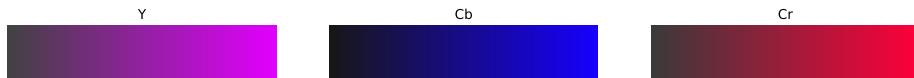


Figura 15: Cortes em $Y=128$, para Cb variando nos 8-bits, Cr também variando nos 8-bits e Y representa Cb e Cr variando nos 8-bits juntos. As matrizes foram convertidas para o formato RGB para poderem representar as cores que cada nível é capaz de produzir.

6. **Pergunta:** Ajustando os parâmetros do monitor há alterações visíveis? Quais? Como? Explique e Justifique!

Sim, se ajustarmos os mesmos parâmetros que antes do monitor as cores se alteram. O monitor deixa de ser capaz de produzir certas cores e por isso as cores das barras se alteram.

1.4 Subamostragem de Cor

1. **Tarefa:** Considere as imagens do item anterior.

- (a) Converta-as para YCbCr.
- (b) Reduza as “resoluções” horizontais e verticais da luminância Y. Para isso, passe antes um filtro de média pela componente Y. Isto é, convolua-a com a matriz

$$\text{Filtro} = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \quad (2)$$

sendo a origem o elemento central. O MatLab disponibiliza a função `conv2` para realizar convoluções bidimensionais.

- (c) Gere as imagens coloridas resultantes com a luminância Y filtrada.
- (d) Faça o mesmo com as componentes de cor Cb e Cr.
- (e) Para cada uma das quatro imagens originais, apresente as imagens resultantes a partir das duas filtragens acima.

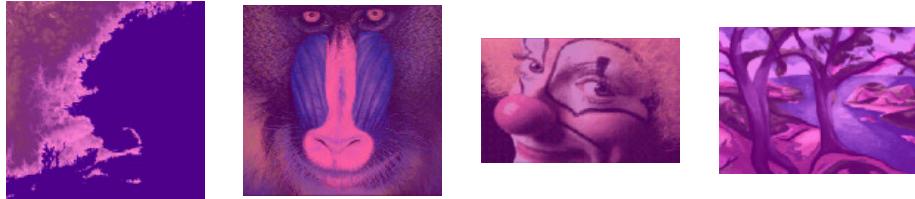


Figura 16: Imagens referentes a filtragem de Y.

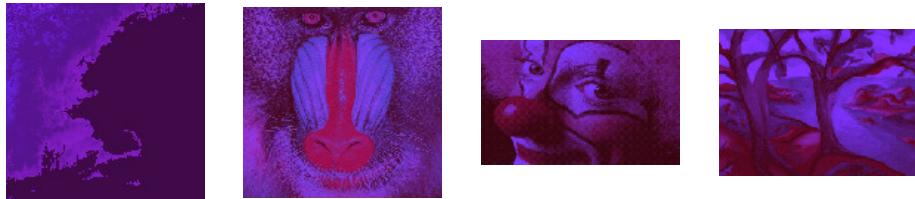


Figura 17: Imagens referentes a filtragem de Cb.

2. Pergunta: Explique o que faz o filtro definido na equação (2).

O filtro faz a varredura dos elementos de cada nível de cor, divide por 9 e os soma; é como um filtro de média bidimensional, eleminando grandes variações entre elementos próximos.

3. Pergunta: Comente os resultados observados no item 1 e explique-os.

O filtro reduziu grandes variações entre os valores de Y para o caso da Figura 16, tornando as cores com uma tonalidade média levemente rosada. Já para os elementos



Figura 18: Imagens referentes a filtragem de Cr.

de Cb, como vemos na Figura 17, os elementos foram todos levados a um tom mais azulado. Enquanto que para os elementos de Cr, observado na Figura 18, os tons da matriz Cr levaram as imagens a tons mais fortemente rosados.