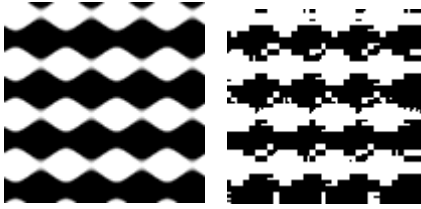


As decisões tomadas para se usar o método foi considerar que, em ambos os métodos, considere que a imagem era uma função contínua e no bicúbico que também era de classe C^2 , mesmo que as imagens dadas não fossem.

Para imagens preto e branco, comprimir e depois descomprimir gera imagens mais deformadas.

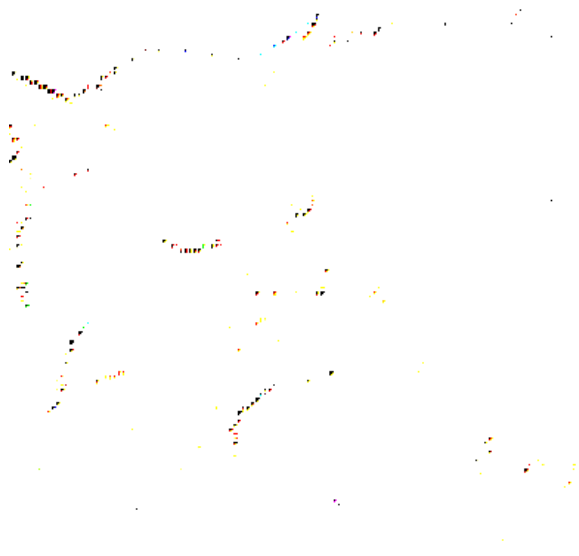


Essas imagens foram geradas com $f(x,y) = 2*\sin(y)+3*\sin(x)$

Para compressão e descompressão, o método deixa as fotos pixeladas independente do k pertencente aos naturais escolhido mas gera imagens que lembram a original (pelo menos com funções que são de classe C^2).

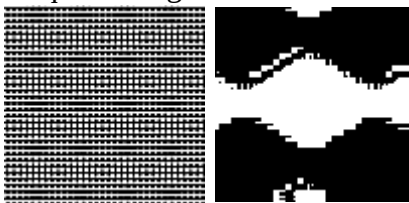
Para funções de classe C^2 , a imagem ainda fica pixelada e cria uma imagem que lembra a original mas com diferenças que já foram faladas antes.

Para funções que não são de classe C^2 , o método não funciona e marca onde fica a transições de cores como nas fotos a seguir:



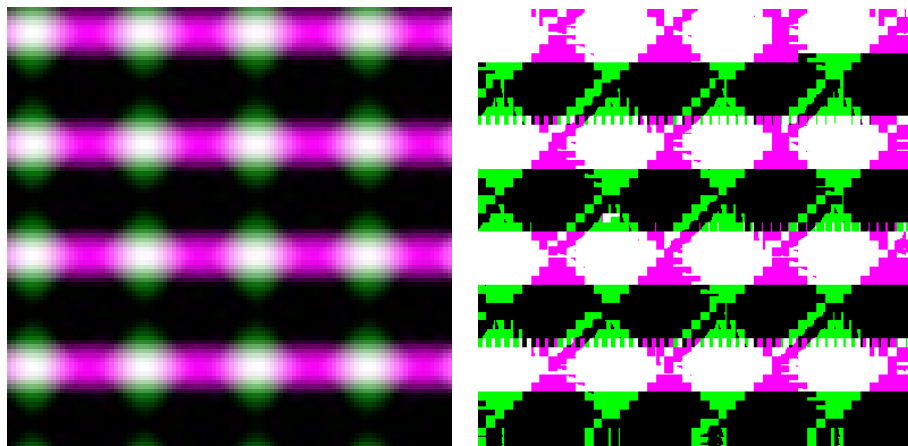
e isso também ocorreu com outras fotos testadas.

Para um $h = 2$, após a compressão e descompressão ocorreu algo que achei muito interessante, que foi que a imagem formada é uma parte dos padrões formados pela imagem original:



já para casos que o h é pequeno, a interpolação se da melhor

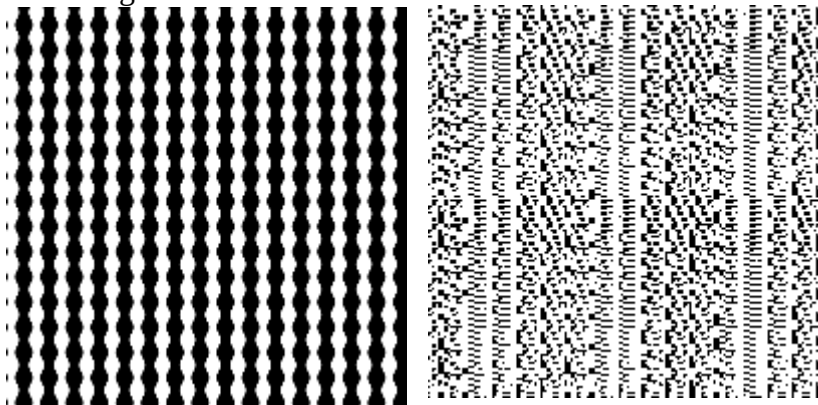
Ao comprimir com $k = 7$ e descomprimir, a imagem ficou um pouco desfigurada, ficando parecida com um zoom da imagem comprimida (mostrando os pixels), além das cores ficarem mais fortes e definidas em relação a imagem original, além de ter algumas distorções, segue os exemplos:



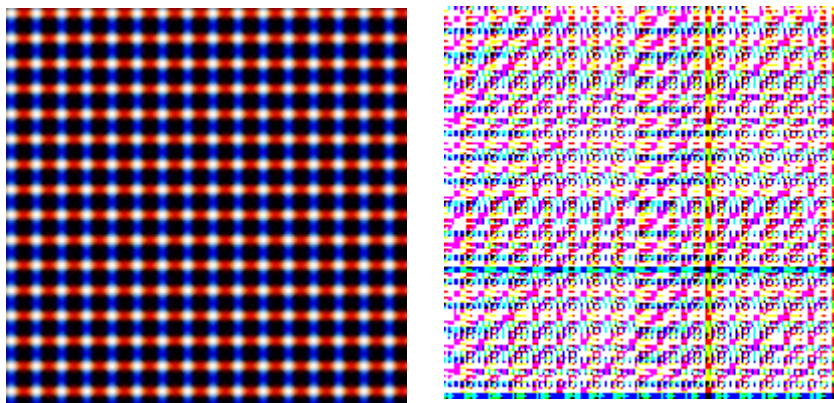
onde inicialmente foram usados $p = 100$.

Bicubico:

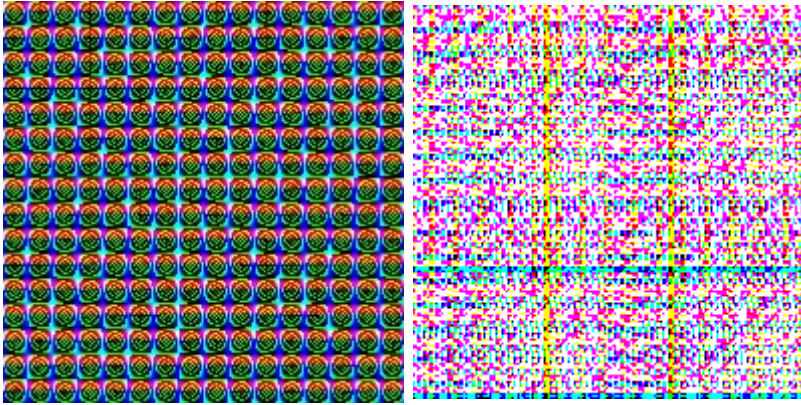
Para imagens em preto e branco o metodo não funciona muito bem mas ainda se parece levemente com a original



Para imagens Coloridas de classe C^2 o metodo não funcionou muito bem após a compressão e depois descompressão, mas a imagem ainda lembra um pouco a imagem original. As fotos a seguir foram os resultados:

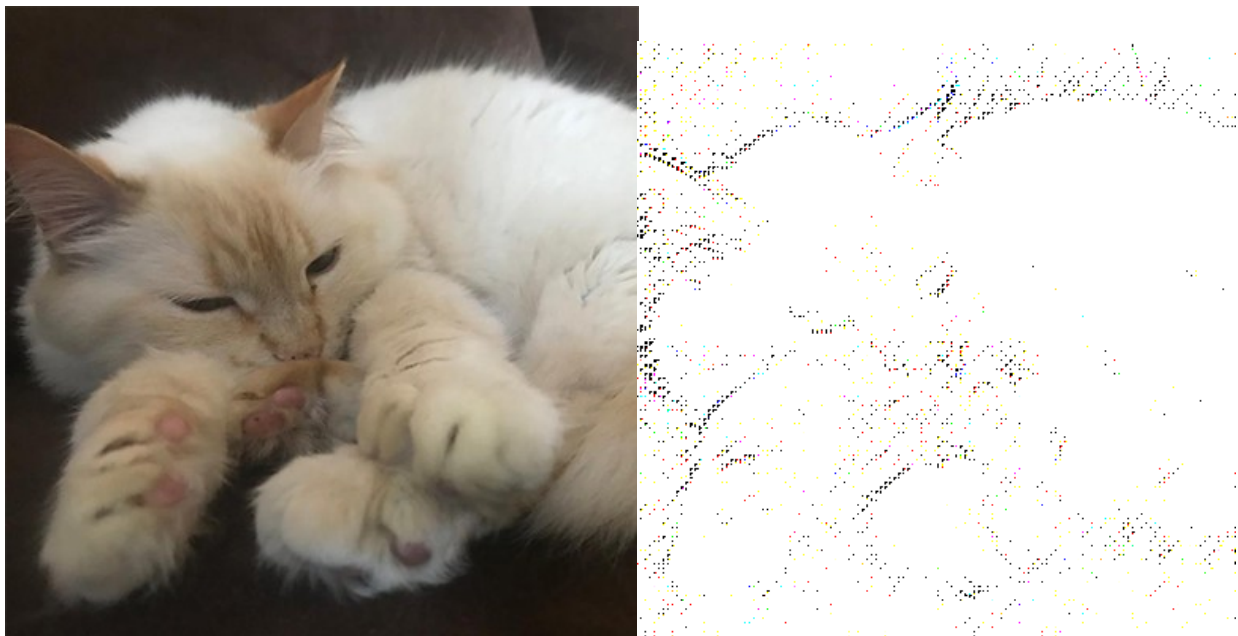


O método bicubico não funciona muito bem para funções de classe C^2



função que gerou a primeira imagem: $f(x,y) = (\sin(x), \tan(x*y), \cos(y))$

Para funções que não são de classe C^2 o método não funciona muito bem também, pois ele apenas é visível os pixels que formam as bordas do desenho da imagem, mas ainda assim é melhor que o primeiro método



Para o método bicubico, usando $k=7$ para a compressão e descompressão criou uma imagem que fica apenas nas bordas de troca de cores da imagem, lembrando em muito pouco (ou quase nada) a imagem original dando o seguinte resultado:

