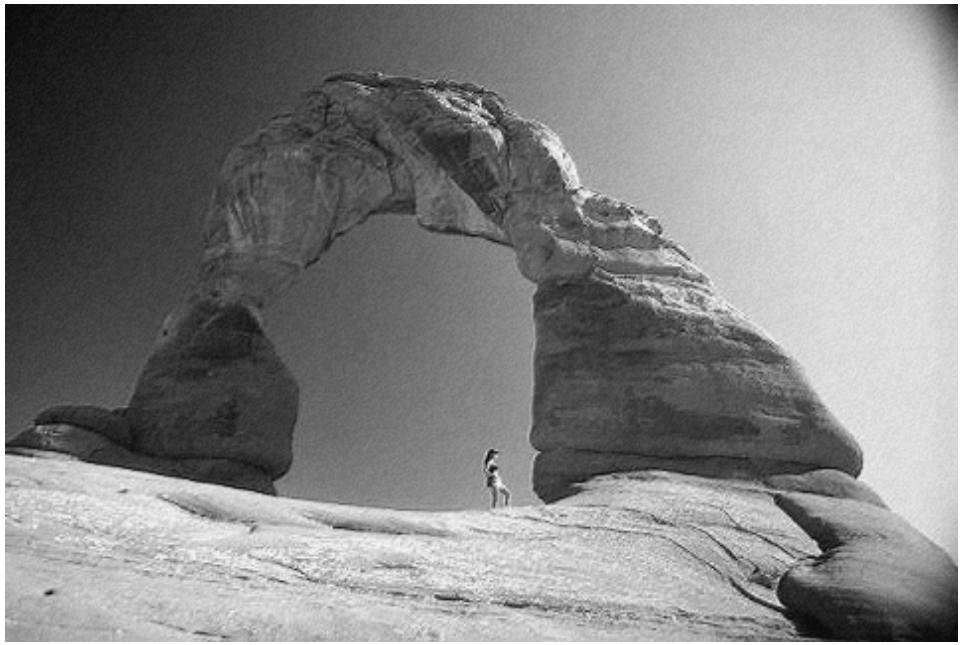
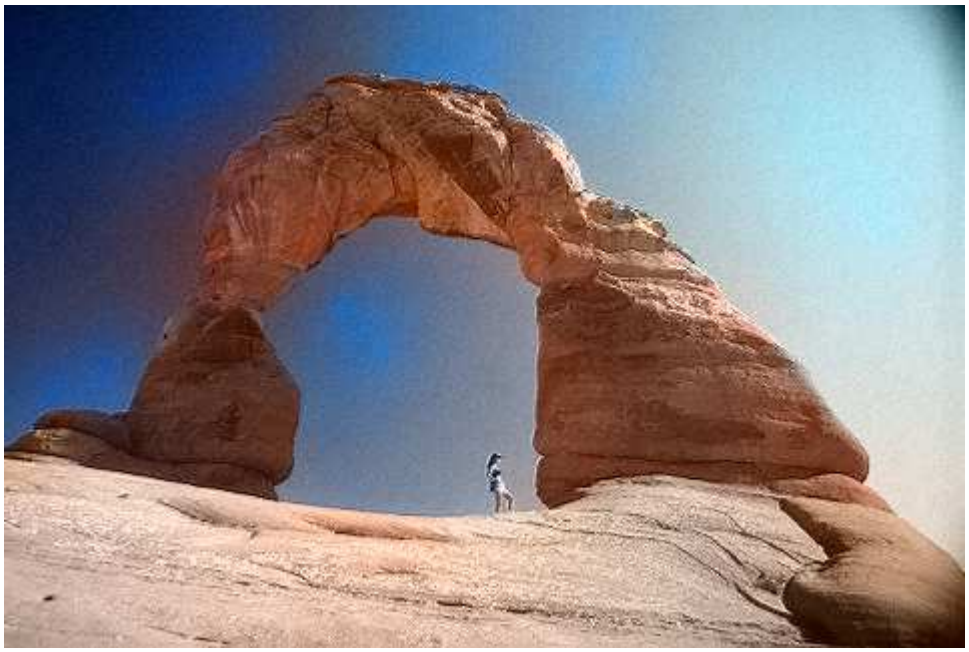
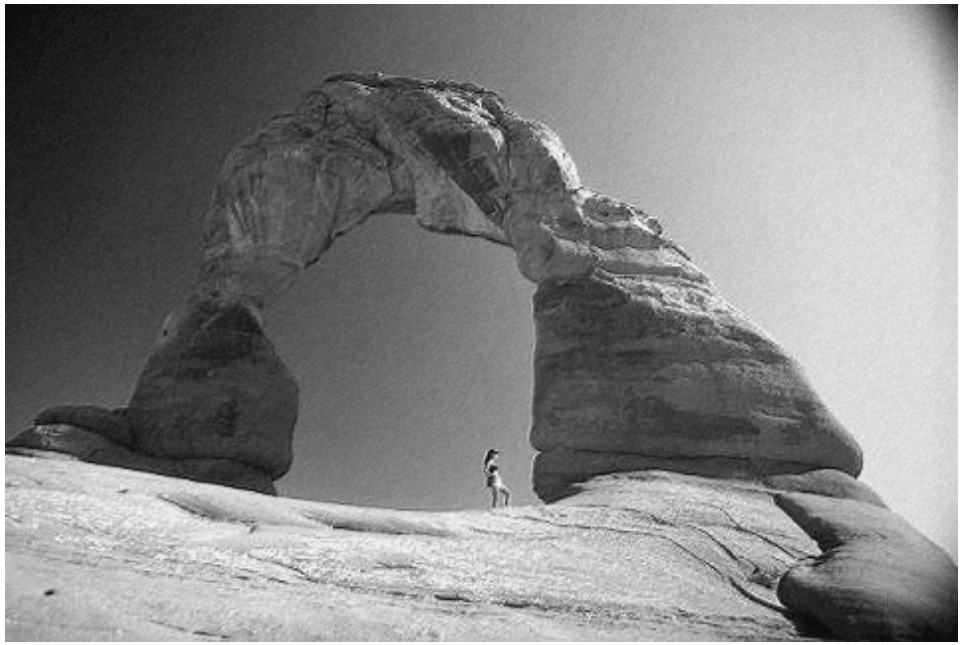


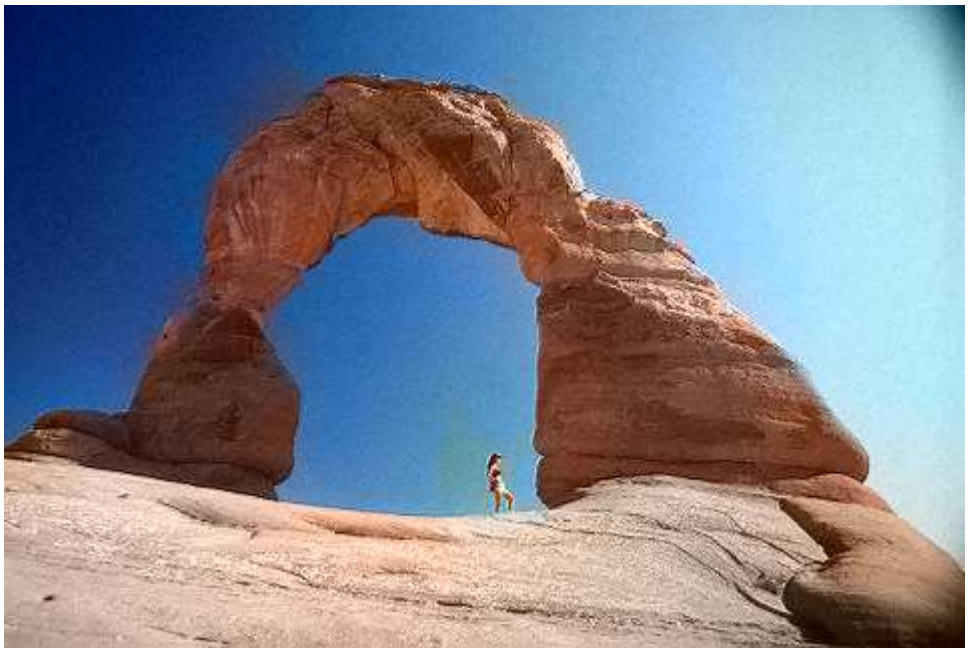
こうなった。





誤差に注意しないと破壊的結果。
誤差なのか拘束条件で指定したピクセルか分からなくなる。拘束条件と判定されるとその周囲の色が動かなくなってしまう。

拘束条件とノイズ（差分）の区別が付かない。
Float -> int にしてBRGの定義域を[0.00,1.00]から[0,255]に変更してやる。
拘束条件が決まるまでは整数演算してやるとノイズは出ない。その後に定義域を[0.00,1.00]に戻す。



計算した画像

オリジナル画像

拘束条件

途中までは比較的短時間で組みあがった。
最終段階で色々とハマッたけど何とか克服。

新たに参考にしたやつ
interpolation by Solving a Minimization Problem (Optimization)
「http://scicomp.stackexchange.com/questions/11387/interpolation-by-solving-a-minimization-problem-optimization」

その他
論文ではYUVにして、、、といっているが論文書いた本人はNTSC（ YIQ形式）に変換している。しかしよく考えると多分どっちでもいい筈。
つまり、輝度を変えずにカラーが弄れるから。根本は白黒画像の輝度は絶対に守るという事だからだと思う。

https://www.reddit.com/r/ColorizedHistory/

- Photoshop職人達が歴史的に有名な白黒写真をカラー加工して投稿し合う「**COLORIZED HISTORY**（カラー化された歴史）」というのがある。
- 作ったプロトタイプで勝負
  ※画像加工は全く経験なしだがツールで勝負

ただし、まだバグバグで間違った結果が沢山でてしまうし理屈が合わない処理もあるのでいい結果だけで勝負している。実際にはまだ計算が正しくないが画像にしてしまうと良く見ないと誤りは見えないが高分解能画像だとバレバレ。

プロの作品との比較

職人の技

計算したヤツ

プロの作品との比較

職人の技

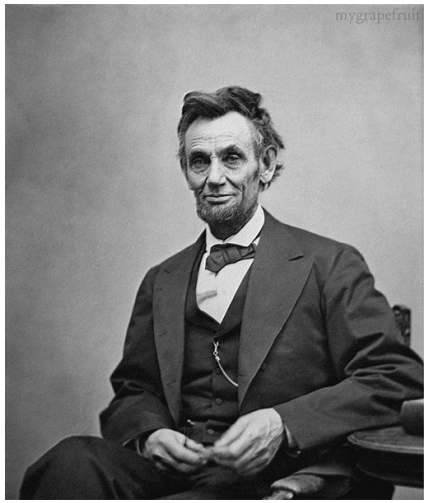
計算したヤツ

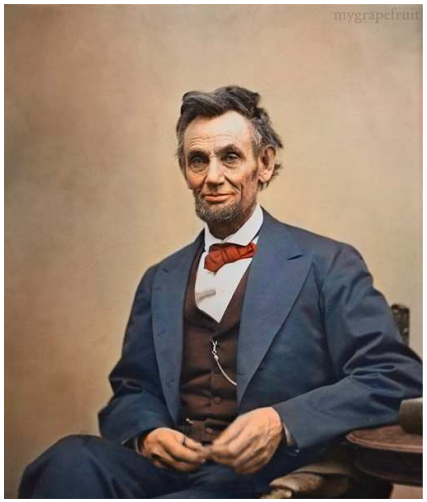
プロの作品との比較

職人の技

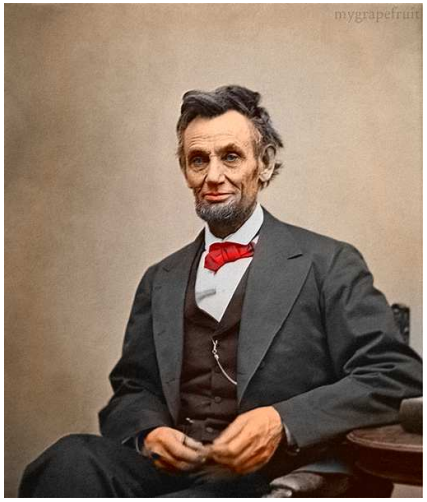
計算したヤツ

プロの作品との比較





職人の技



計算したヤツ

スーツの色は違う色
になるように計算。