

Improving scans of black and white photographs by recovering the print maker's artistic intent

Daniel M. German

dmg@uvic.ca
Dept. of Computer Science
University of Victoria.

Abstract

In this paper we propose a method that reverse engineers the aesthetic decisions made by a print maker to produce a print from a negative, namely cropping, contrast selection, and dodging-and-burning. It then re-applies this process to the electronic negative in order to achieve an electronic version of such print with better tonal range and detail than one produced by scanning the print. We then extend this method to restore a print by combining scans of different versions of the same image.

1. Introduction

Institutions and individuals are scanning their inventories of photographs, both negatives and prints. The most valuable negatives of a collection have usually been printed. It is not uncommon for a scan from a negative to look different than the scan of a print. This is often because prints (particularly those created in chemical darkrooms) were hand crafted, done by print makers (darkroom operators) who applied their skills and knowledge to try to bring the best of a negative. Negative scanning is preferred to print scanning because more detail can be recovered from the negative than from the print. An operator is likely capable of turning a negative scan into a version that is very close to the look of the print. But this requires skill and time. Automatic scanning is desirable due to its considerable saving in time and money. It is desirable to automatically recover and reuse the operations done by the print maker to reapply them to the negative to obtain a version that is as close as possible to the negative, but with the richness in tone and detail that the negative provides.

1.1. Print making

Ansel Adams, one of the most influential photographers of the 20th century once stated that “the negative is the score, and the print is the performance” [Rud96]. Photographs are rarely recognized via their negatives; what we recognized are their prints. Ansel Adams wrote “we start with the nega-

tive [...] and proceed through a series of ‘work’ prints to our ultimate objective, the fine print” [Ada95]. Given a negative, it is sometimes desirable to recover a particular “fine print” as it was envisioned by its creator (not just any print, even an excellent one).

Often overlooked, the artistic decisions made by the print maker are what frequently set apart such fine prints. Some photographers perceive printing as an integral part of their art. Brett Weston (son of Edward Weston) was notable for destroying most of his negatives when he turned 80 years old so nobody else would ever print them. His views were probably affected by his belief that his brother Cole was making inferior prints from their father’s negatives [Cla03]. Photographers like Brett Weston believe that there are only a small number of valid performances for a negative.

Print making is a craft. It requires a good understanding of the technical aspects of the printing process, an artistic eye that can apply such techniques to a negative to bring the best of it, and plenty of practice. Photographic printing merges art and craft to express an idea or a feeling that has been captured in a negative [Rud96].

In this paper we explore how to computationally recover four of the most important aesthetic decisions that a print maker has already made to create a specific print [Rud96] in order to reapply this information to the electronic negative to create an electronic copy of the print with its rich detail and a look that attempts to match the paper print:

Cropping Many photographers object to having their negatives cropped and expect their photographs to always be printed with the edge of the negative (most notably Henri Cartier-Bresson). When a negative is cropped is usually for one of many reasons: a) restrictions of the paper to use (for example, photo papers are rarely found in sizes proportional to 35mm negatives), b) for aesthetic reasons: to balance the image, to emphasize an element in it, or to exclude areas, c) to correct framing, such as a tilted negative, and d) to change the message of a photograph (by excluding information that might change the opinion of the viewer). Cropping is very subjective.

Determine the shadows (blacks), highlights (whites) and midtones (mid grays) of the print. Automatic printing and scanning often results in images that look too dark or too light. The perfect brightness of a print is highly subjective, and depends on both the contents of the image (i.e. snow vs a dark room), and the desired feeling that it should convey.

Determine the contrast (tonality) of the print. The contrast of a print changes its mood and there are no rules with respect to what contrast brings the best from a print. Many photographers believe that every print should include almost any tone of gray from pure blacks and pure whites (such as those using the Zone System developed by Adams [Ada95]), while others are take a totally opposite position and produce images with only pure white and black. The contrast of a print is usually determined by the type of paper used and how it is developed (its characteristic contrast curve).

Dodge-and-burn. A print from a negative might require that some of its regions be lighter or darker. This is achieved by selectively under exposing (dodging) or over exposing (burning) such regions. A print that has not been dodged-and-burned is said to be a straight print. Determining the regions that should be burned or dodged is also a very subjective decision.

Scanning a negative and applying automatic midpoint and contrast adjustment produces the equivalent of a straight print, similar to those made by an automatic photo minilab: the scanning software has no knowledge of the contents of the negative and simply finds the white and black points, and applies a predetermined contrast function. These scans will look different from prints that have been cropped or dodged-and-burned.

In this paper we propose a method that uses the detail available in a high resolution digital scan of the negative, and combines the cropping, midtones, contrast, and dodging-and-burning of one of its prints (potentially of significantly lower resolution and level of detail) to create a superior copy of the negative with its rich detail and a look that attempts to match the print. We extend this method to make it possible to combine two images in such a way that their sum is better than either one. We illustrate this method with several photographs.

1.2. Related Work

Few algorithms have been designed to automatically crop an image based upon its content [Sob06, KDR02, SAD*06] but with little regard to their aesthetic value. Geigel and Musgrave modeled the film development process and their goal was to create digital images that had the “look” of film. [GM97] Reinhard et al. proposed a method to automatically tone and dodge-and-burn a HDR image to a LDR; their method is based on the method used by Ansel Adams (the zone system) and attempts to provide a faithful reproduction of the original [RSSF02]. Bae et al. proposed a method to transfer the tonality of one image to another [BPD06] but it does not take into consideration dodge-and-burn. The SIFT algorithm (proposed by Lowe [Low04]) and remapping algorithms for image correction and compositing (such as those used in the creation of panoramic images, see [Sze06] for an overview) allow matching of images of different resolutions and levels of detail.

2. Model

Let us define *neg* as the negative image, and *print* to a print originating from the same negative. The process of producing *print* can be modelled as follows:

$$print \equiv paperF(projNeg, t) \quad (1)$$

$$projNeg \equiv scaleCrop(proj(neg, t)) \quad (2)$$

paperF is the contrast function of the paper used. *projNeg* is the projection of the negative projected by the enlarger on the paper (*proj*), scaled and cropped (*scaleCrop*).

The image of the negative projected by the enlarger (*projNeg*) is affected by the enlarger itself: the intensity of the light it produces, the optical characteristics of the lens (such as its optical distortion, the aperture used to make the print), the geometric distortion produced when the plane of the film and the plane of the papers are not perfectly parallel. Also, if the negative bulges (something common in large negatives that are not held in place with glass carriers) it will also distort the projected image. This image is then cropped and scaled to the desired size. Cropping can be done at two different stages: in the negative carrier, and at the paper level (using an easel). When the cropping is done in the negative carrier, the edges of material used to crop the negative interact with the negative itself, resulting in a unique pattern along the edge of the printed image. *paperF* will depend upon the type of developer and temperature used.

The projected image is projected on the paper for a length of time, the paper is developed (for another length of time), resulting in a print.

Let us assume that cropping is done at the paper level (the image is first enlarged, then cropped), therefore:

$$scaleCrop \equiv crop \circ scale$$

We can assume that the projection of the negative does not affect the contrast function of the paper, nor vice-versa: thus:

$$print \equiv crop(scale(proj(paperF(neg,t)))) \quad (3)$$

We are not concerned with recovering the time of exposure nor the paper's contrast function, but rather, the effective contrast function given a particular time of exposure (in other words, we want the shifted curve as it was applied to the negative for this particular print), hence:

$$print \equiv crop(scale(proj(contrast(neg))))$$

We define the `invertedPrint` as the uncropped, unscaled, unprojected print:

$$invertedPrint \equiv proj^{-1}(scale^{-1}(crop^{-1}(print)))$$

Thus:

$$contrast(neg) = invertedPrint$$

`proj`, `crop`, and `scale` (and their inverse) can be estimated using the registration and remapping algorithms typically used in panorama making. The `invertedPrint` will be an approximation to the negative (due to light diffusion of the enlarger, quantization errors in scaling, difficulties to perfectly match the distortion of the lens, etc.). As a consequence it is very difficult to match the pixels on the negative to the `invertedPrint` in order to estimate `contrast`. Instead, we take advantage of a property that the histogram of the `invertedPrint` should be the same as the histogram of `contrast(neg)`:

$$hist(contrast(neg)) = hist(invertedPrint)$$

`contrast` can be estimated using a histogram matching algorithm. We can then match the original print from the negative using:

$$matchPrint = crop(scale(proj(contrast(neg)))) \quad (4)$$

`contrast` can also be estimated using a more advanced algorithm, such as the two-scale tone matching [BPD06] (Bae's method suffers from JPEG artifacts that histogram matching does not). Each of the four functions `crop`, `scale`, `proj` and `contrast` have been approximated. Fortunately their errors are relatively independent of each other: `scale` and `crop` operate on the geometrical boundaries of the projected image, `proj` depends on the relative position of each pixel with respect to each other. `contrast` depends on the tonality of the original print with respect to the negative and can be approximated even if the pixels do not match perfectly.

2.1. Algorithm for Print Matching

Equation 4 is the basis for our print matching algorithm, depicted in Figure 2 and illustrated in Figure 1. It can be briefly described as follows: it takes a black and white negative and a print to match (they can be of different sizes), and an indication of whether the negative should be cropped to match the print. The first step is to register and remap the print

to match the negative's size (the SIFT algorithm is used to match the two images even if the images differ significantly in contrast and size [Low04], remapping is done using `panotools` (<http://panotools.sourceforge.net>). At this point the cropping is determined, and the contrast function computed using the intersection of both images. The contrast function is applied to the negative, and cropped if necessary: the result is the matched print.

func `matchPrint(neg, base, crop) ≡`

Pre-conditions:

`neg` and `base` have

brightness normalized to [0,1]

and each might have an alpha channel

indicating regions to be ignored.

`crop` is a flag that indicates if result should be cropped.

Post-conditions:

`matched` contains the reproduced print

with brightness normalized to [0,1]

begin

`(basemapped, cropping) =`

`RegisterRemapAndScaleBase(neg, base);`

`croppedNeg = cropping(neg);`

`fc = MatchHistogramFunction(basemapped, croppedNeg);`

if `crop` **then**

`matched = fc(croppedNeg);`

else

`matched = fc(neg);`

fi

end

Figure 2: Algorithm to match a base print. It takes two parameters: the inverted scanned negative (`neg`) and the print to match (`base`). The function `RegisterRemapAndScaleBase` is responsible for the registration and remapping of the base print so it matches as close as possible its corresponding region in the negative.

Figure 3 shows an example of the use of the algorithm. The original negative looks relatively flat. The base print is scaled and remapped to match the negative, and the cropping of the negative is determined. The contrast function is calculated (shown in Figure 4) and applied to the negative, which is then cropped. The tonality of the new print closely matches the one of the base print. There is more detail in the clouds, and in the dark areas (such as her dress, under the coat), details that are lost in the print scan. In this example the cropping is important: it has straighten the print, and cropped part of the sky.

The `matchPrint` algorithm is useful even when only smaller sections of the original image are used as a base print, as long as such region has a good tonality range (in this case no cropping is computed). Also, the algorithm performs reasonably well when two prints are used as input (it is always preferable to use a negative scan, but that is not

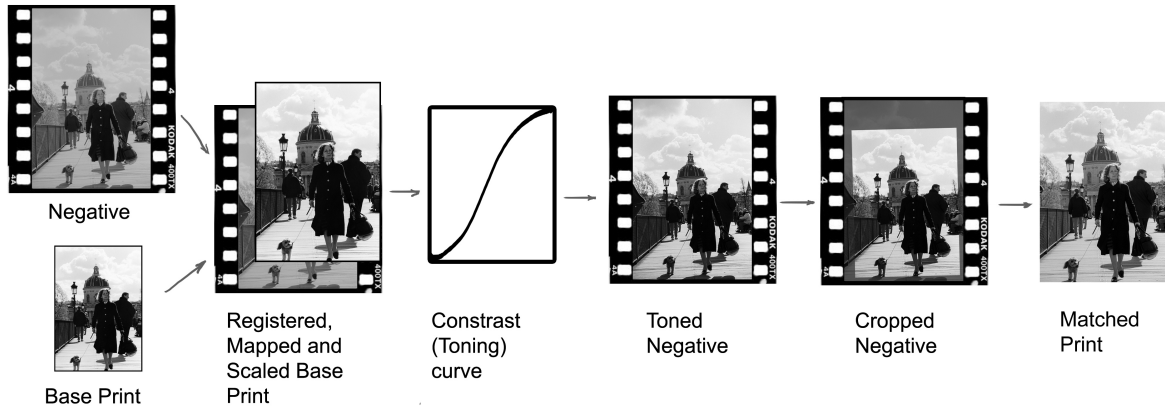


Figure 1: Diagram illustrating the major steps of the `matchPrint` algorithm.



Figure 3: Crossing Pont Neuf by D.M.German, used with permission. The first image is the negative and looks flat. The second is the base print and shows lack of detail in the shadows (i.e. dress under her coat) and highlights (areas of the sky). The third shows the computed cropping, when applied to the negative. The last image is the matched print. Notice how the shadows and highlights have been recovered, while maintaining the overall contrast of the base print.



Figure 5: Sadie Pfeifer, 48 inches high has worked half a year. (1908) by Lewis Hine; in the public domain. An example of a matched print created using only a section of the original image as the base print. The image on the left corresponds to a scan of a print by the Library of Congress (nclc 01455). The one in the middle is a scan from a book [Kot05] (a double page spread, hence the darker area in the middle; it has been slightly blurred to reduced the effect of halftoning printing). Only the boxed area is used as the base print. The one on the right is the matched print.

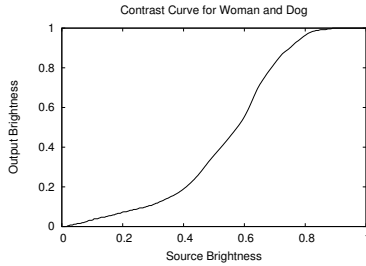


Figure 4: Computed contrast curve for Crossing Pont Neuf in figure 3.

always possible). Figure 5 shows the use of `matchPrint` to a scan of a gelatin print by Lewis Hine and a base print that corresponds to a small section of the scan (from a book).

2.2. Restoring photographs by combining images

Sometimes it is necessary to restore a small section of a print, and there exist other versions of the same image that can be used to replace such damage. The `matchPrint` algorithm can be used to match a section of one version to the section of the other, so they can be combined. This process is described in Figure 6. It takes two input versions of the same image (it can be easily extended to take n input versions) and produces a restored one. The quality of the restoration will primarily depend upon two factors: the first one is how well the tonality of the area to use matches the tonality of the area to be restored; the second is how well the detail in the area to use matches the detail in the area to be restored and is primarily affected by the relative difference in size between both images and the level of detail present in them.

Figure 7 demonstrates the use of this algorithm. The Library of Congress scan of *Carolina Girl* (Figure 5) has what appears to be water damage in the left-bottom corner. The scan from [Kot05] is used to restore the depicted area. The `restore` algorithm performs well in this case because the area to restore is away from the center of attention of the original image, and it is not in sharp focus. Figure 8 shows another example, where a damaged and badly scanned print, but of high resolution is matched to a lower resolution scan from a book.

3. Dodging and Burning

Dodging and burning involves selectively projecting areas of the negative for a longer or shorter time, respectively. Using equation 3 we can model a dodged and burned print as:

$$DBprint \equiv crop(scale(proj(paperF(neg), T)))$$

where T is a matrix of the same size as neg indicating the projection time for each pixel. Without loss of generality, let

```
funct Restore(base, i)  $\equiv$ 
```

```
Pre-conditions:
```

```
base is the best of the available images
i is another version of base that overlaps in
regions that need to be restored
```

```
Post-conditions:
```

```
restored contains the restored image
```

```
begin
```

```
// compute an alpha channel of the region to be restored
```

```
// and apply it to i
```

```
ic = selectRegionToRestore(base, i);
```

```
bm = matchPrint(base, ic, TRUE);
```

```
restored = blend(base, bm);
```

```
end
```

Figure 6: Algorithm to restore an image *base* using image *i*. `selectRegionToRestore` could be implemented automatically by first matching both prints (using `matchPrint`) and then computing areas in which they differ significantly; or manually, with the assistance of an operator. `blend` takes both the matched region to restore, and the base print, and combines them without a visible seam (it can be implemented using an algorithm such as [Aga07]).



Figure 7: The scan from the Library of Congress has water damage in the bottom left corner (see Figure 5). Fortunately it is an area of low detail. A region from the scan from [Kot05] (depicted also in Figure 5) is used to restore it.

us assume that there is no cropping or scaling:

$$DBprint \equiv proj(paperF(neg), T)$$

$paperF$ is increasingly monotonic, hence we rewrite this equation as the sum of a print at any reasonable time t , plus a dodge-and-burn mask contribution for each pixel (negative for dodging, positive for burning):

$$DBprint \equiv proj(paperF(neg, t)) + dbMask$$

The `matchPrint` algorithm can be used as an estimation of the middle term in the previous equation using the dodged-and-burned base print $DBprint_0$:

$$print_m = matchPrint(neg, DBprint_0)$$



Figure 8: Dust Storm, Cimarron Country, 1937, by Arthur Rothstein; in the public domain. Left, high resolution scan from dirty gelatin print, Library of Congress, USA (ppmsc.00241); center scan from book [Ros84] crediting a gelatin print also from the Library of Congress, USA; right, restored image. For this photo the sky of the match print has been replaced with the sky from the base print and has been slightly cropped to remove the edge imperfections.

$$DBprint \equiv print_m + dbMask \quad (5)$$

Thus:

$$dbMask = DBprint_0 - print_m$$

Unfortunately $DBprint_0$ and neg are unlikely to be perfectly aligned (due to quantization errors, optical projection, and scanning distortions—if they were perfectly aligned $DBprint$ would be equal to $DBprint_0$). We address this problem by blurring both sides of the previous equation:

$$blur(dbMask) = blur(DBprint_0 - print_m) \quad (6)$$

Any blurring algorithm can be used as long as it does not significantly affect too much the contrast of the overall image; the level of blurring will depend on the size of the image. The goal is to diffuse any harsh edges present in the difference of the two images, so their luminance is comparable. We need to unblur $blur(dbMask)$. To do so we multiply it by the original negative and match its contrast by computing the function f_c s.t:

$$hist(blur(dbMask)) = hist(f_c(blur(dbMask) * neg))$$

resulting in:

$$blur(dbMask) = f_c(blur(dbMask) * neg)$$

which we can combine with 5 and 6 which results in:

$$dbMask \equiv \text{matchPrint}(neg, DBprint_0) + f_c(blur(DBprint_0 - print_m) * neg) \quad (7)$$

3.1. Algorithm to recover dodging and burning

Equation 7 becomes the foundation for our algorithm to reproduce a dodged-and-burned print. In order to generalize it is necessary to crop the negative (if necessary), and scale and distort the base print to match the negative as close as

possible (using the `matchPrint` algorithm described in section 2.1). The algorithm is described in Figure 9: its two parameters are the negative, and the dodged-and-burned print to match.

func *MatchDodgedAndBurnedPrint*(*neg*, *base*) \equiv

Pre-conditions:

neg and *base* have
brightness normalized to [0,1]

Post-conditions:

print contains the reproduced print
cropped to match *base*

begin

```
// scale base to match neg, and
// determine how neg should be cropped
(scaledBase, cropping) =
  RegisterRemapAndScaleBase(neg, base);
// match neg as close as possible to base
matched = MatchPrint(neg, base);
matchedb = blur(matched);
maskb = blur(scaledBase) - matchedb;
// maskb can have values in [-1,1]
t = cropping(neg) * maskedb;
fc = MatchHistogramFunction(maskb, t);
mask = fc(t);
print = matched + mask;
```

end

Figure 9: Algorithm to reproduce a dodged and burned print. It takes two parameters: the inverted scanned negative (*neg*) and the print to match (*base*); *base* and *neg* do not need to have the same size, and if necessary *neg* will be cropped to match *base*.

We exemplify our algorithm using the photograph *Roadside Stand Near Birmingham* by Walker Evans (1936). The scanned negative originates from the Library of Congress (5145x4173 pixels, including margin), the base print is a reduced version of a print manually created by John Hill (Yale School of Art) and was published electronically by the New York Times [Kim06] (531x450 pixels). Figure 10 shows the negative scan, the base print, the matched print, and the burned-and-dodged match print. The base print was created from this scan of the negative using a digital process [Kim06]. This print has very localized dodging and burning, and it is depicted in Figure 11. For example, the areas outside the store has been burned. The contrast of the outside has also been increased, without affecting the brightness nor the contrast of the inside (the face of the girl and the man are similar in both versions). The outside areas behind the left window and door are not altered either, but the outside area behind the right window has been burned.

We have found that adding directly the burned-and-dodged mask to the negative tends to reduce the contrast of burned or dodged areas. We have experimented with



Figure 10: Roadside Stand Near Birmingham by Walker Evans (1936). Negative from the Library of Congress; in the public domain. Its printed area is approx 4800x3780 pixels. The second image is the base print: a low resolution version by John Hill as it appeared in [Kim06]. It measures 531x450 pixels and was created digitally from the same scan of the negative. The cropping properly centers and balances the print. Notice how the matched print is still very light in the foreground. The fourth one is the burned-and-dodged matched print; it measures 4303x3679 pixels. The burned areas in the foreground have lost contrast. The last one uses burn-and-dodging with scaling. It produces a better image and it is very close to the base print.

other methods to dodge and burn and obtained better results when the same level of burning applies more to darker areas than lighter ones (and vice-versa for dodging). We call this method dodging-and-burning with scaling. An example of such method is Photoshop’s “color dodging” and “color burning”. Because this dodging and burning tends to have a lesser effect compared to the additive method, it is necessary to compensate by darkening the masks before they are applied.

Figure 12 shows the application of the Match-DodgedAndBurnedPrint algorithm to *Migrant Mother* by Dorothea Lange. Notice how the base print shows burning in the lower area (below the baby’s body). In this case there is no significant dodging in the image.

4. Conclusions and Further Work

The main disadvantage of the algorithms presented in this paper is that they require two or more copies of the same image (one of them, preferably, its negative).

The information recovered using `matchPrint` could be very useful for historians, who can use it to compare multiple variants of the same image. For example, we have found 5 different croppings of *Migrant Mother*, including one that covers almost all the negatives. Also, when we created a match print based on an early print of *Migrant Mother*, and compute the difference between both images it is obvious how Lange altered the negative (as described in [Kot05]).

With respect to future work. We believe the toning of the match print can be improved by using algorithms that take into consideration only the regions around a given pixel (such as [BPD06]). Similarly, the dodge-and-burn algorithm presented here needs to be improved to avoid reducing the contrast of the dodged or burned areas. It is necessary to automatically determine the size of the blurring kernels, and what are the best parameters for the application of scaled dodging-and-burning. It is also possible for a hybrid method to be the most effective: either one in which a human operator determines such parameters for each negative; or one in

which an operator uses the output of these method (particularly the contrast curve and the dodge-and-burn mask) and extends them for a better result.

These algorithms can also be extended to take into account color, both in black and white (toning) and color images. The problems are similar: the scan and the print “look” different, but one wants to recover the printer’s intent, to be able to match the colour negative to the colour print.

Finally, it is necessary to formally evaluate the results of this method, and compare them against other automatic methods (such as a direct application of Bae et al. two-scale toning method [BPD06]).

References

[Ada95] ADAMS A.: *The Print*. Bulfinch, 1995.

[Aga07] AGARWALA A.: Efficient gradient-domain compositing using quadtrees. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers* (2007), p. 94.

[BPD06] BAE S., PARIS S., DURAND F.: Two-scale tone management for photographic look. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers* (2006), pp. 637–645.

[Cla03] CLAIR J. S.: A Natural Eye. The Photography of Brett Weston. In *Counterpunch*, Cockburn A., Clair J. S., (Eds.). Dec 2003.

[GM97] GEIGEL J., MUSGRAVE F. K.: A model for simulating the photographic development process on digital images. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (1997), pp. 135–142.

[Joh06] JOHNSTON M.: Print Offer: Dorothea Lange’s *Migrant Mother*. The Online Photographer. http://theonlinephotographer.blogspot.com/2006/12/print-offer-dorothea-langes-migrant_15.html, Dec 2006. Accessed March 2008.

[KDR02] KIMBELL B. D., DALTON D. L., RUDD M. L.: System and method for automatically cropping graphical



Figure 11: Dodging (left) and burning (right) applied to the matched print (blurred and unblurred). The dodged version is presented as the absolute value of pixels below zero, and the burned as the positive values only. Notice in particular how the burning applies mainly to the foreground and the two kids, while the dodging tries to brighten the rest of the print. For presentation purposes the pixels in these masks have been inverted: white means no contribution, and black full contribution

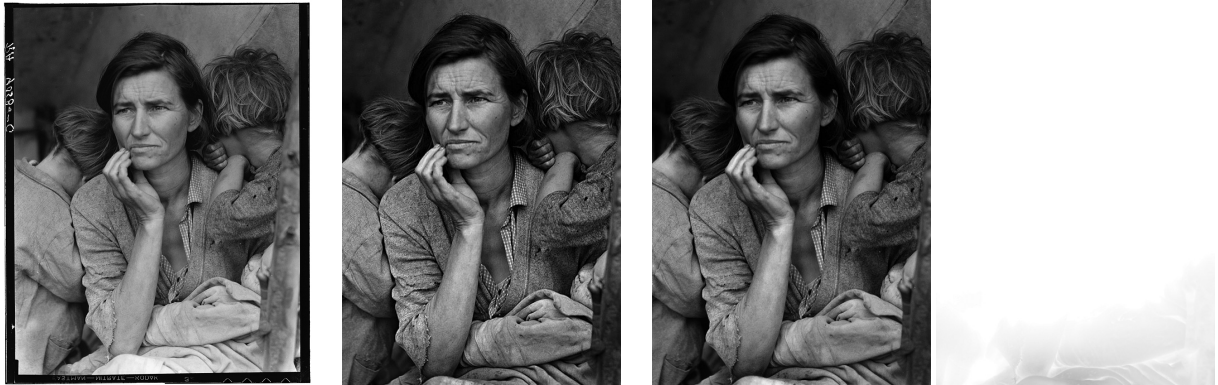


Figure 12: Migrant Mother, 1936, by Dorothea Lange; in the public domain. The image corresponds to the scan of the 4x5 negative done by the Library of Congress of the United States (LC-DIG-fsa-8b29516). The base print is significantly lower resolution and was created by Mike Johnston [Joh06]. The third is the dodged-and-burned matched print (using the scaling method). The final image is the unblurred burned mask. Notice how the bottom of the image has been darkened. The image required no significant dodging.

images. United States Patent 6456732, Sept 2002. Assignee Hewlett-Packard Company.

- [Kim06] KIMMELMAN M.: Walker Evans. Or Is It? The New York Times, Art Review, August 25 2006.
- [Kot05] KOTZLE H.-M. (Ed.): *Photo Icons. The story behind the pictures*. Taschen, 2005.
- [Low04] LOWE D. G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision* 60, 2 (2004), 91–110.
- [Ros84] ROSENBLUM N.: *A World History of Photography*. Abbeville Press, New York, USA, 1984.
- [RSSF02] REINHARD E., STARK M., SHIRLEY P., FERWERDA J.: Photographic tone reproduction for digital images. In *SIGGRAPH '02: Proc. of the 29th annual conference on Computer graphics and interactive techniques* (2002), pp. 267–276.
- [Rud96] RUDMAN T.: *The Photographer's Master Printing Course*, 2nd ed. Reed International Books LTD, 1996.
- [SAD*06] SANTELLA A., AGRAWALA M., DECARLO D., SALESIN D., COHEN M.: Gaze-based interaction for semi-automatic photo cropping. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems* (New York, NY, USA, 2006), ACM, pp. 771–780.
- [Sob06] SOBOL R. E.: System and method for automatically cropping graphical images. United States Patent 7034848, April 2006. Assignee Xerox Corp.
- [Sze06] SZELISKI R.: Image alignment and stitching: a tutorial. *Found. Trends. Comput. Graph. Vis.* 2, 1 (2006), 1–104.