

Extracting the essence from sets of images

Thanks to the author for permission to use this paper on the Research Skills course. Copyright, the Eurographics Association. This copy of the paper is provided under the University of Cambridge's license from the Copyright Licensing Agency. Copying this copy is a breach of copyright.

Abstract

We use a set of photographs showing similar scenes as a model for a single photograph of this scene. A distance measure for this model is defined by correlating the neighborhoods of pixels in similar positions. A cross analysis of the source images yields confidence values for their pixels. The confidence values together with the distances in pixels are used to steer a variable bandwidth mean shift algorithm that moves an arbitrary image towards one conforming with the model. Furthermore, distances are also used for a non-local means reconstruction of image areas that have no consistent explanation in the source images. This allows reconstructing images of scenes that are inconsistently documented in the source images, e.g. are occluded in the majority of images.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture and Image Generation

1. Introduction

We are tackling the problem of synthesizing new images from a set of photographs showing a similar scene, inspired by photographic art that is based on blending many single photographs to reveal patterns or regularities. The technical challenges appear similarly in the area of Computational Photography [RTLN07], where a set of photographs from the same viewpoint are combined to overcome limitations of the classical photographic process, or achieve certain artistic effects. Examples include:

- Combining several illumination situations [Hae84].
- Blending together panoramas from different images [Che95, SS97]
- Generating and then mapping images with increased dynamic range from a set of digital photographs [DM97, Ash02, FLW02, RSSF02, DD02, MMS05]
- Removing artifacts of flash lighting by combination with a photograph of the naturally lit scene [ARNL05, PSA*04, ED04]
- Combining several instances of a moving object [FZ03]

More examples and a detailed discussion can be found in the notes of courses on Computational Photography [RTLN07] and on the websites of its authors. In many of these works it is assumed that the scene is constant while all images are taken. However, in all but studio situations this is not the

case: many objects move and appear in different positions or in all but few of the images. This problem has been considered especially in the area of high dynamic range imaging, as *de-ghosting* (see Chapter 4.7 of the book by Reinhard et al. [RWPD05] and in particular [KAR06] for an approach that is very related to ours), it could also be dealt with interactively and using graph cuts [ADA*04].

We are aiming at unsupervised learning of the likely scene elements from a set of images. While we demonstrate our basic algorithm at the example of image sets taken from the same viewpoint, our main goal is synthesizing new images from sets of images showing different objects, inspired by works in photography based on blending many input photographs. Compared to simply averaging or supervised blending, our synthesis process is based on a model that is supposed to distill the “essence” from the input automatically. We will later explain how our process differs from related approaches.

If the set of images has a clearly dominant value for each pixel it is likely that these dominant values represent the constant part of the scene and they could be easily extracted (such as with standard de-ghosting approaches). We have found, however, that in many of the situations we are interested in only a subset of the pixels give rise to an easily identifiable static image content. Still, as long as the static



Figure 1: Seven photographs of the Reichstag building in Berlin taken within 5 minutes. The left image shows the whole photograph, the other images show a smaller part for the 6 other images.

content is visible in a (possibly small) subset of images we would like extract it.



Figure 2: The mean of the source images (top) is smooth but exhibits ghosting artifacts, and the flags are smeared out. The image of pixel values with maximum likelihood (middle) as well as or forming large clusters (bottom) are discontinuous, including seemingly smooth areas such as the sky.

See for example the set of images in Figure 1, which have been taken during five minutes with the camera placed on an object. While it appears that this set would be easy to deal with, several approaches based on statistics of corresponding pixels fail (Figure 2): simply taking the mean value for each pixel generates "ghosts" of transient image content, as expected. Maximizing the likelihood for each pixel, or converging to the largest cluster of similar pixel values generates discontinuous results over smoothly varying areas, here mostly visible in the sky. Note that because of these slight differences in tone and brightness we do not want to select pixels from the source images. Yet, this is what graph cut approaches would do, which is why they are likely to fail on the data we are dealing with.

Our main idea for improving the results of these simple approaches is rather optimizing spatial coherence in the image domain then coherence over the set of source image pixels. In particular, we prefer pixels that have identical neighborhoods in a few images over pixels with similar values in many images. Enforcing coherency in the image domain and

might also help inferring information for regions in which the source images have no simple statistics. The results for the technique we have derived based on this principle for the example images mentioned before are shown in Figure 3.

Our approach to implementing these goal combines several tools from image processing. We implicitly define a model from the image sequence by a distance measure (section 2), which compares pixels based on their (relative) neighborhood values. However, distance to the source images based on the mean neighborhood distances, is averaged over only a subset of the source images to accommodate transient elements in the source images. This distance measure is then exploited to steer a variable bandwidth mean shift algorithm (section 3). Distances control both weighting the values in the source images and kernel sizes. If this local mean shift fails to converge for some regions we switch to a non-local version, filling in regions of the image with parts that are consistent with the model (section 4).

Applications of this procedure are manifold. A particular use is to exploit other artistic computational photography techniques based on images sets. We are more aiming at generating new, aesthetically pleasing images, which are blurry only in areas that have no dominant explanation in the input, and clearly depict content that appears in several images.

2. The model

The main idea of this work is that a set of discrete images $\mathcal{I}_j, j \in 1, \dots, n$ describes the scene. We refer to the pixel at (x, y) in the j -th image as $\mathcal{I}_j(x, y)$. Furthermore, we assume throughout this paper that all images have identical dimensions.

For our example, we assume that similar areas in the images are associated with similar scene elements. For the examples with the same scene we have the input images using [War03].

The *model* is implicitly described by the source images. First, we define the distance of an image to the model. Based on this definition of distance, we compute the distance of each of the source images to the model. This yields confidence values for each pixel in the source images, which could be used to refine the model.



Figure 4: The confidence images for the set of input images for the Reichstag set of images, computed based on 7×7 neighborhood patches. Darker pixels reflect small distance to the model represented by the image set and, thus, high confidence. Conversely, light pixels depict low confidence. Only the first five out of seven source images are shown.

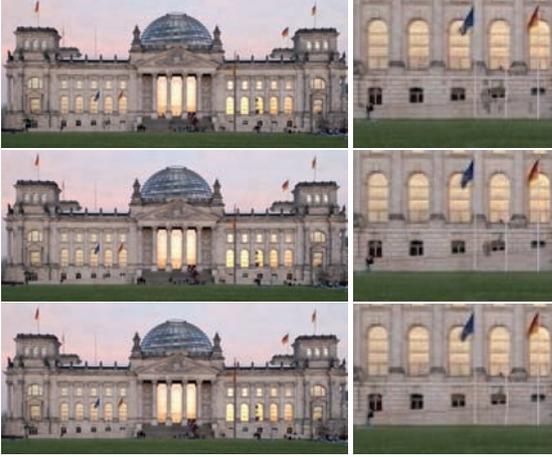


Figure 3: Results of the mean shift procedure we are proposing to move images towards the model associated with the set of source images. Here, coherence is optimized over neighborhoods of different sizes – from top to bottom: 3×3 , 5×5 , 7×7 .

2.1. Distance to the model

Given a test image \mathcal{T} , we like to quantify for each pixel $\mathcal{T}(x, y)$, how likely it is to be consistent with the model described by the source images $\{\mathcal{I}_j\}$. For a pixel to be consistent with the model it should be similar to k out of the n source images.

However, this classification should be spatially coherent in the test image space, i.e. if isolated pixels in \mathcal{T} belonging to a transient object happen to be consistent with the model they should not be classified as consistent. We achieve this spatial coherency by comparing pixel neighborhoods rather than only the pixel.

So, let $p_{\mathcal{T}}(x, y)$ be a patch around (x, y) , i.e. the vector

$$\begin{aligned} p_{\mathcal{T}}(x, y) = & (\mathcal{T}(x-s, y-s), \dots, \mathcal{T}(x+s, y-s), \\ & \dots, \\ & \mathcal{T}(x-s, y+s), \dots, \mathcal{T}(x+s, y+s)), \end{aligned} \quad (1)$$

containing the pixel values of a $(2s+1) \times (2s+1)$ sub-image around (x, y) .

In order to evaluate $\mathcal{T}(x, y)$ we compare $p_{\mathcal{T}}(x, y)$ with all $p_{\mathcal{I}_j}(x, y)$. Specifically, if the illumination in all source and test images is similar, we simply use Euclidean distance of the patch vectors as the measure of their difference, i.e.

$$d_{\mathcal{I}_j, \mathcal{T}}(x, y) = \|p_{\mathcal{T}}(x, y) - p_{\mathcal{I}_j}(x, y)\| \quad (2)$$

Alternatively, if illumination might change over source images or is expected to be different in the test image, we use individually normalized patch vectors with zero mean, i.e.

$$\hat{p}_{\mathcal{T}}(x, y) = \frac{p_{\mathcal{T}}(x, y) - \bar{p}_{\mathcal{T}}(x, y)}{\|p_{\mathcal{T}}(x, y) - \bar{p}_{\mathcal{T}}(x, y)\|} \quad (3)$$

where $\bar{p}_{\mathcal{T}}(x, y)$ is a constant vector of the mean of $p_{\mathcal{T}}(x, y)$. We denote the differences of these vectors as $\hat{d}_{\mathcal{I}_j, \mathcal{T}}(x, y)$. Note that this distance is similar in spirit to normalized cross correlation, the classic building block in many visual feature analysis algorithms [DHS00].

The k -smallest individual difference values are then averaged to yield to the final distance to the model, i.e.

$$\begin{aligned} \mathcal{D}_{\mathcal{I}, \mathcal{T}}(x, y) &= k^{-1} \sum_{c=1}^k d_{\mathcal{I}_{j_c}, \mathcal{T}}(x, y), \\ d_{\mathcal{I}_{j_l}, \mathcal{T}}(x, y) &\leq d_{\mathcal{I}_{j_{l+1}}, \mathcal{T}}(x, y), l \in [1, \dots, n-1]. \end{aligned} \quad (4)$$

The main reasoning for using only the k -smallest individual distances is to compensate for the fact that not all source images are representative for the scene content at (x, y) . A close fit to only k of all n source images is significant and means that the test image is very close to the model at this point. Note that k need not be greater than $n/2$ for this approach to work – in fact, we have used $k = 3$ or $k = 4$ for most of the examples.

2.2. Confidence values for the source images

Using the operator \mathcal{D} we compute for each source image \mathcal{I}_j a confidence image \mathcal{C}_j . In particular, \mathcal{C}_j is defined as $\mathcal{D}_{\mathcal{I}, \mathcal{I}_j}$, however, with k increased by one, as \mathcal{I} contains \mathcal{I}_j and the distance of an image to itself is zero. Figure 4 shows the set of confidence images for the Reichstag example.



Figure 5: The convergence process and the adapting kernel sizes for a set of 7 images of the Siegessäule and a street with heavy traffic in Berlin. Here, we have used neighborhood patches of size 5×5 .

3. Mean shifting towards the model

The main idea for moving a template image \mathcal{T} towards the model is to shift each pixel value towards a weighted mean of pixels in the source images, i.e. \mathcal{T} is replaced by \mathcal{T}' , repeatedly, where \mathcal{T}' is defined as

$$\mathcal{T}'(x, y) = \sum_{j=1}^n \frac{w_j(x, y)}{W(x, y)} \mathcal{I}_j(x, y) \quad (5)$$

where the weights $w_j(x, y)$ are computed locally and based on the distance of \mathcal{T} to each of the source images \mathcal{I}_j as well as the confidence maps. This is inspired by mean shift algorithms [CRM01, CM02], which have proved to be useful in mode and feature selection (e.g., image segmentation).

The natural ingredient for the weights $w_j(x, y)$ are the distance $d_{\mathcal{T}, \mathcal{I}_j}(x, y)$ of the current template pixel $\mathcal{T}(x, y)$ to each of the source pixels $\mathcal{I}_j(x, y)$ and the confidence of each of the source pixels $\mathcal{C}_j(x, y)$. We use a Gaussian for defining the weight based on these distance, i.e.

$$w_j(x, y) = e^{-\frac{d_{\mathcal{T}, \mathcal{I}_j}^2(x, y)}{h^2(x, y)}} e^{-\frac{\mathcal{C}_j^2(x, y)}{h^2(x, y)}} \quad (6)$$

$$W(x, y) = \sum_{l=1}^n w_l(x, y)$$

One key idea in our mean shift algorithm exploits the convenient fact that the kernel size $h(x, y)$ can be selected locally and adaptively during the mean shift iterations [Com03]. The reasoning for estimating the kernel size is as follows: if the distance $\mathcal{D}_{\mathcal{T}, \mathcal{I}}(x, y)$ in a pixel to the model is large, we need to use a large kernel; however, if the distance is small, then using a smaller kernel will favor a subset of pixel values in the source images that are apparently representing static scene content. Consequently, we set

$$h(x, y) = \mathcal{D}_{\mathcal{T}, \mathcal{I}}(x, y). \quad (7)$$

The general idea of adjusting the individual contributions

of the source images for each pixel is similar to the deghosting approach of Khan et al. [KAR06]. However, our definition of contributions is different as it mostly depends on the k most similar neighborhoods in the source images. Notice what this definition entails for pixels with a dominant value among the source images: as the value in the template moves towards the dominant value, the distance to this value get smaller and so does the kernel. This further increases the relative weights for the dominant value, further reducing the kernel, and so on, until the kernel size is zero and the pixel assumes the dominant value. In this way, our adaptive bandwidth mean shift algorithm removes outliers in the source images and converges to the dominant value (see Figure 5 for a sequence of mean shift steps together with gray scale coded images of the distances/kernels). In contrast, the algorithm of Khan et al. always considers all source pixels, and its stable state would usually include, albeit with small weights, also the outlier pixels.

Because distances are taken from image neighborhoods rather than just pixel values, this convergent behavior is spatially coherent, i.e. if the source images contain several dominant modes, mode selection is done coherently over the template. This makes the algorithm stable even for pixels that contain the static scene content in only very few source images. In other words, sub-dominant modes in the model could be selected based on spatial coherence.

If a pixel has no dominant value in the source images whatsoever, the kernel size will not vanish during the mean shift iterations. This is typically the case for pixels that also have low confidence $\mathcal{C}_j(x, y)$ for all source images, i.e. pixels for which there is no easily extractable information. However, the non-vanishing kernel during the iteration is very useful information, namely reflecting the fact that the pixel value is not consistent with the model.



Figure 6: A set of images showing the Brandenburg gate in Berlin, taken within 10 minutes, while the camera has similar but not completely identical view directions.

4. Non-local mean shifting

While the mean shift algorithm quickly converges to dominant pixel values in the source images, it cannot provide a useful answer for pixels that have no clear explanation in the model (see the set of images in Figure 6, especially the area under the Brandenburg gate detailed in Figure 7 and the results for the mean shift procedure in Figure 8). Our idea to overcome this problem is to use values of pixels with high confidence elsewhere in the image. The idea of placing pixels based on neighborhood statistics is common for stochastic textures [EL99, WL00, EF01], however also works for structured content, i.e. filling holes in images [DCOY03]. Our approach is mostly inspired by smoothing using non-local means [BCM05a, BCM05b] and in terms of the resulting algorithm very similar to the UINTA approach by Awate et al. [AW05, ATW05].

If the mean shift iterations yield a constant non-zero kernel size, we are extending the comparison of patches and convex combination of pixel values to a small neighborhood around the current pixel (x, y) . However, performing this extended search for good matching pixel neighborhoods in source images can be very expensive. We exploit the fact that usually many pixels in the template image did converge to a dominant pixel value. So it is sufficient to search over pixels in the template having small distance to the model.

In particular, given a template image T' after few mean shift iterations, we perform a non-local mean shift for pixels $T'(x, y)$ with $\mathcal{D}_{T', \mathcal{I}}(x, y) > \epsilon$ as follows:

$$T''(x, y) = \sum_{u=x-s}^{x+s} \sum_{v=y-s}^{y+s} \frac{\omega(x, y, u, v)}{\Omega(x, y)} T'(u, v) \quad (8)$$

Here, the weights $\omega(x, y, u, v)$ are based on the distances $\mathcal{D}_{T', \mathcal{I}}(u, v)$ and an adapted distance metric for comparing patches within T' . This adapted distance metric takes the confidence $\mathcal{D}_{T', \mathcal{I}}$ into account, i.e. distances between values are considered to be important only for pixels that are believed to have reasonable value. In particular, rather than computing the inner product of the difference vectors $p_{T'}(x, y) - p_{T'}(u, v)$ with itself and then the square root,



Figure 7: Zoom-in of the pedestrian area under the gate, showing that the walls of the gate are occluded in all but a few images. Also note that the colors of the transient objects (clothing of the pedestrians) is very similar due to the illumination.

we simply use

$$t(x, y, u, v) = \frac{\left\langle p_{T'}(x, y) - p_{T'}(u, v), p_{\mathcal{D}_{T', \mathcal{I}}}(x, y) \right\rangle^{1/2}}{\left\| p_{\mathcal{D}_{T', \mathcal{I}}}(x, y) \right\|} \quad (9)$$

as $\mathcal{D}_{T', \mathcal{I}}(x, y)$ only contains non-negative values.

Based on this distance measure for comparing patches within T' we define the weights, here using the distances

from \mathcal{T}' to the model as confidence values

$$\omega(x, y, u, v) = e^{-\frac{r^2(x, y, u, v)}{h^2(x, y)}} e^{-\frac{\mathcal{D}_{\mathcal{T}', \mathcal{T}}^2(u, v)}{\varepsilon^2}}$$

$$\Omega(x, y) = \sum_{u=x-s}^{x+s} \sum_{v=y-s}^{y+s} \omega(x, y, u, v). \quad (10)$$

Note that we use ε as kernel size for the confidence term – the reason is that we want to give high weights only to pixel values that are close to the model.

We have found it particularly useful to repeat the standard mean shift procedure after few iterations of this non-local smoothing, however, with k decreased (see Figure 9). What happens is that the non-local smoothing fills in trusted pixel values into regions with untrusted values. These values might be found in the source images, yet perhaps only few of them. Reducing k allows pixel values to converge against smaller clusters of values. While this would be dangerous in the beginning of the process as pixel values could converge against sub-dominant values, when most pixels are already stable this only affects untrusted pixels.

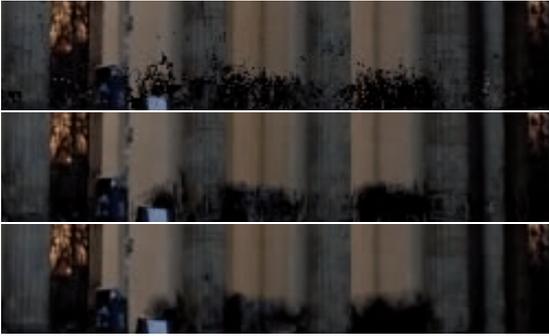


Figure 8: The three rows show the results of the mean shift procedure with patch sizes 1×1 (similar to per pixel cluster analysis), 3×3 , and 5×5 . Because of the incoherent input the output also fails to resemble the gate.

5. Application and results

We have implemented the operators and the mean shift procedures as they are described above using the CImg library. Results can be found in the images used as illustrations in the previous sections. In all our results we have used neighborhood patches with zero mean, but have not normalized the magnitude.

A step of the mean shift procedure requires considering each pixel and its neighborhood in each of the source image. So, quite naturally, the computational complexity of this procedure linearly depends on the number of pixels in the input images, the number of images, and the number of pixels in the neighborhood. From our experiment we suspect that the constant involved is heavily affected by cache coherency,

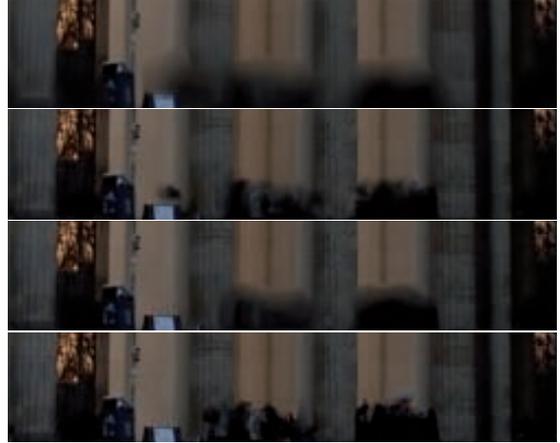


Figure 9: Applying the non-local mean filter and the apply mean shift to the result: the first row shows the non-local means result starting from the 5×5 mean shift results depicted in the last row in Figure 8. For the next row this result has been used as the starting point of mean shifting. The following two rows show another application of the same sequence.



Figure 10: The full image of one the results obtained for the image sequence depicted in Figure 6. We have obtained a variety of interesting results, showing that different parameter settings might give better results for particular parts of the scene, but it is difficult to get the expected result in all parts for one set of parameters.

which we have not optimized. It also appears natural to keep track of the pixels that are still moving towards the mean in each step. We have also not yet exploited this information.

Apart from applying the procedures to images of exactly the same scene, especially when using the non-local mean shift approach we can also process images of different objects. We have applied the technique to a set of images of very similar, yet different content: Figure 11 shows a set of photographs of five *different* benches (observe the different



Figure 11: Images of 5 different benches. While the benches might look similar, the concrete in the background exhibits different patterns. Note that no attempt was made to register the images.

patterns of the concrete around the bench). As the benches, the geometry of the background, and the viewpoints are slightly different we have not registered the images – we assume the non-local process will still find static scene elements. Indeed, interweaving mean shift and non-local means we are able to get interesting results (Figure 12).

We have also experimented with adding all transient objects to an image rather than trying to remove them. However, it turned out to be very difficult to avoid occlusions that are clearly improper.

6. Discussion

The mean shift procedure based on patch neighborhoods proved to be simple and very useful. It could be used as a basic tool in other environments, such as panoramic image composition or HDR. In comparison to other, similar approaches such as Khan et al. [KAR06] we find that it is possible to converge to exactly the static scene content even if this content is visible only in a small subset of the input images, by exploiting coherence in image domain.

All results had been obtained by starting from the mean input image. However, if k is chosen to be very small, the starting point is important for the mean shift procedure, as the process tends to converge to the closest cluster of similar neighborhoods. Using other starting points we had been able to generate "better" results for some of the inputs, however, in a sense that made the approach semi-automatic rather than fully automatic.

The idea of non-local means for image improvement worked in a few examples, however, we feel that it is generally only mildly interesting for applications in computational photography (for example, Figure 10 shows one example of what is possible with this approach, however, we are sure better results could be obtained by further tweaking the parameters). It already produces interesting results when applied to the input we find more interesting: photographs of different objects, so necessarily also from different viewpoints and with little chance that registration could be of any help (see Figures 11 and 12). Nevertheless, we believe a better approach is to really search in all input images in a non-local way. Yet, this requires a good implementation that avoids the nested loops resulting from a straightforward implementation.



Figure 12: A result obtained for the unregistered photographs of the five different benches using a combination mean shift on 7×7 neighborhood patches and non-local means.

One important addition is the appropriate selection of pixels values with normalized patch distances: while it is easy to use normalized distances, it is unclear what values to use in the mean shift algorithm. So far we have used the values in the source images, however, it would make more sense to apply the mean shift algorithm to the normalized neighborhood values and the reconstruct an image from the normalized neighborhoods. We are currently working in this direction.

References

- [ADA*04] AGARWALA A., DONTCHEVA M., AGRAWALA M., DRUCKER S., COLBURN A., CURLESS B., SALESIN D., COHEN M.: Interactive digital photomontage. *ACM Transactions on Graphics* 23, 3 (Aug. 2004), 294–302.
- [ARNL05] AGRAWAL A., RASKAR R., NAYAR S. K., LI Y.: Removing photography artifacts using gradient projection and flash-exposure sampling. *ACM Transactions on Graphics* 24, 3 (Aug. 2005), 828–835.
- [Ash02] ASHIKHMEN M.: A tone mapping algorithm for high contrast images. In *Rendering Techniques 2002*:

- 13th Eurographics Workshop on Rendering* (June 2002), pp. 145–156.
- [ATW05] AWATE S., TASDIZEN T., WHITAKER R.: Unsupervised texture segmentation with nonparametric neighborhood statistics'. In *European Conference on Computer Vision* (2005), pp. 494–507.
- [AW05] AWATE S., WHITAKER R.: Unsupervised, information-theoretic, adaptive image filtering for image restoration. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 28, 3 (2005), 364–376.
- [BCM05a] BUADES A., COLL B., MOREL J.-M.: A non-local algorithm for image denoising. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 60–65.
- [BCM05b] BUADES A., COLL B., MOREL J.-M.: A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation* 4, 2 (2005), 490–530.
- [Che95] CHEN S. E.: Quicktime vr - an image-based approach to virtual environment navigation. In *Proceedings of SIGGRAPH 95* (Aug. 1995), Computer Graphics Proceedings, Annual Conference Series, pp. 29–38.
- [CM02] COMANICIU D., MEER P.: Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24, 5 (2002), 603–619.
- [Com03] COMANICIU D.: An algorithm for data-driven bandwidth selection. *IEEE Trans. Pattern Anal. Mach. Intell.* 25, 2 (2003), 281–288.
- [CRM01] COMANICIU D., RAMESH V., MEER P.: The variable bandwidth mean shift and data-driven scale selection. In *ICCV* (2001), pp. 438–445.
- [DCOY03] DRORI I., COHEN-OR D., YESHURUN H.: Fragment-based image completion. *ACM Transactions on Graphics* 22, 3 (July 2003), 303–312.
- [DD02] DURAND F., DORSEY J.: Fast bilateral filtering for the display of high-dynamic-range images. *ACM Transactions on Graphics* 21, 3 (July 2002), 257–266.
- [DHS00] DUDA R. O., HART P. E., STORK D. G.: *Pattern Classification*. Wiley-Interscience Publication, 2000.
- [DM97] DEBEVEC P. E., MALIK J.: Recovering high dynamic range radiance maps from photographs. In *Proceedings of SIGGRAPH 97* (Aug. 1997), Computer Graphics Proceedings, Annual Conference Series, pp. 369–378.
- [ED04] EISEMANN E., DURAND F.: Flash photography enhancement via intrinsic relighting. *ACM Transactions on Graphics* 23, 3 (Aug. 2004), 673–678.
- [EF01] EFROS A. A., FREEMAN W. T.: Image quilting for texture synthesis and transfer. In *Proceedings of ACM SIGGRAPH 2001* (Aug. 2001), Computer Graphics Proceedings, Annual Conference Series, pp. 341–346.
- [EL99] EFROS A. A., LEUNG T. K.: Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision* (1999), pp. 1033–1038.
- [FLW02] FATTAL R., LISCHINSKI D., WERMAN M.: Gradient domain high dynamic range compression. *ACM Transactions on Graphics* 21, 3 (July 2002), 249–256.
- [FZ03] FREEMAN W. T., ZHANG H.: Shape-time photography. In *2003 Conference on Computer Vision and Pattern Recognition (CVPR 2003)* (June 2003), pp. 151–157.
- [Hae84] HAEBERLI P.: *Grafica obscura*, 1984. web site <http://www.sgi.com/grafica/>.
- [KAR06] KHAN E. A., AKYUZ A. O., REINHARD E.: Ghost removal in high dynamic range images. In *IEEE International Conference on Image Processing* (2006), pp. 2005–2008.
- [MMS05] MANTIUK R., MYSZKOWSKI K., SEIDEL H.-P.: A perceptual framework for contrast processing of high dynamic range images. In *APGV 2005* (Aug. 2005), pp. 87–94.
- [PSA*04] PETSCHNIG G., SZELISKI R., AGRAWALA M., COHEN M., HOPPE H., TOYAMA K.: Digital photography with flash and no-flash image pairs. *ACM Transactions on Graphics* 23, 3 (Aug. 2004), 664–672.
- [RSSF02] REINHARD E., STARK M., SHIRLEY P. S., FERWERDA J.: Photographic tone reproduction for digital images. *ACM Transactions on Graphics* 21, 3 (July 2002), 267–276.
- [RTLN07] RASKAR R., TUMBLIN J., LEVOY M., NAYAR S.: *Computational photography*, 2007. SIGGRAPH 2007 Course.
- [RWP05] REINHARD E., WARD G., PATTANAİK S., DEBEVEC P.: *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting*. Morgan Kaufmann, 2005.
- [SS97] SZELISKI R., SHUM H.-Y.: Creating full view panoramic mosaics and environment maps. In *Proceedings of SIGGRAPH 97* (Aug. 1997), Computer Graphics Proceedings, Annual Conference Series, pp. 251–258.
- [War03] WARD G.: Fast, robust image registration for compositing high dynamic range photographs from hand-held exposures. *Journal of Graphics Tools* 8, 2 (2003), 17–30.
- [WL00] WEI L.-Y., LEVOY M.: Fast texture synthesis using tree-structured vector quantization. In *Proceedings of ACM SIGGRAPH 2000* (July 2000), Computer Graphics Proceedings, Annual Conference Series, pp. 479–488.