Master of Science in Informatics at Grenoble
Master Mathématiques Informatique - spécialité Informatique
option Graphics, Vision and Robotics

Image Blending using Local Phase

Ayush Tewari

26.06.2015

Research project performed at GraphDeco, INRIA Sophia Antipolis

Under the supervision of:
Adrien Bousseau and George Drettakis, INRIA Sophia Antipolis

Defended before a jury composed of:
Dr James Crowley
Dr Edmond Boyer
Dr Nabil Layaida
Dr Dominique Vaufreydaz / Dr J S Franco

[June] 2015

Abstract

Image blending is an important tool for many applications in Computer Graphics and Image Processing. The standard approach to blend images is to take their weighted mean. This approach produces artifacts like ghosting edges when the images are not well aligned. We propose an efficient method to align images using their local phase information, which in turn improves blending. Local phase difference has been shown to represent small motion or shifts between images and has been used for motion manipulation. We study the local phase signals and their difference and propose a method which can extend the limit of shifts they can represent. We test our approach on different images and compare the results with optical flow and a phase-based method concurrent to ours. We finally suggest how our method could be used to improve applications in image-based rendering and image denoising.

Contents

— 1 —
Introduction

Image Blending refers to the use of information from multiple images to create an output image. For example, linear interpolation between two images is a form of blending. In Computer Graphics, image based rendering [Buehler et al., 2001; Eisemann et al., 2008] generates output from new camera positions by blending images captured from nearby input cameras. Image blending is also an important step for some denoising methods [Buades et al., 2005] as well as texture synthesis approaches [Wexler et al., 2004] where similar small patches of any image are blended to generate new patches.

The most common approach to blend images is to linearly combine them i.e. linear interpolation. The image between two images is computed as their mean. If the images are not aligned properly, the edges and structures are repeated multiple times in the output. These artifacts are called the ghosting artifacts (Fig:1.1a). The usual solution is to use optical flow to compute the shifts between the inputs. These shifts can then be used to align the input images before blending them. Since optical flow involves solving a large optimization problem at different scales or some global matching to compute these shifts [Brox et al., 2004; Tao et al., 2012], it is quite slow. Because of the color consistency assumption, optical flow approaches are not robust to drastic changes in illuminance between the images.

(a) Artifacts because of misaligned images in image-based rendering

(b) Sharper image using optical flow[Eisemann et al., 2008]

Figure 1.1: Image Blending Artifacts

Recently, local phase information of images has been shown to be useful for fast local motion/shift manipulations [Wadhwa et al., 2013; Didyk et al., 2013; Meyer et al., 2015]. Phase

differences between two images have been shown to be capable of representing small shifts accurately [Zhang et al., 2015] while multi-scale approaches have been used to represent larger shifts [Didyk et al., 2013; Meyer et al., 2015]. Since phase is invariant to illuminance, phase-based methods perform better than other optical flow based approaches which assume a illuminance consistency between the images. Even though it is possible to shift images just by manipulating their local phases, the range of shifts represented well by the phase difference is very restricted. As the shifts between the images increase, the quality of results degrade drastically. The properties of local phase differences and the issues in representing shifts using them have not been studied extensively.

In this thesis, we analyze the structure and properties of local phase, phase differences and the ambiguities in representing shifts using them. We propose a method to refine the phase differences such that the shifts are represented better than the original phase differences. These refinements can be important for any method which uses phase difference to manipulate images. We also analyze the multi-scale regularization approach to represent larger shifts and come up with an algorithm which can align images before blending them. We consider the input images to be the same signal shifted by different amounts. We build our method by analyzing the case of blending two images. If I and I_{shift} are the two shifted images, we want to find the image I_{middle} which lies between them. Throughout this work, we only consider the problem of interpolating between the images. We also extend our approach to blend more than two images.

— 2 —
Related Work

2.1 Motivating Applications

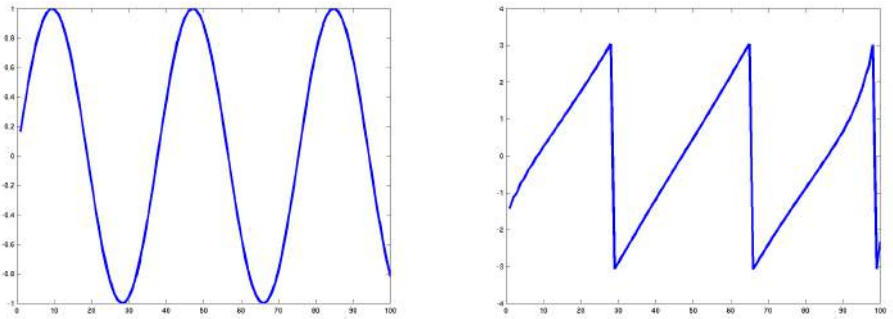
In the context of Image based rendering, Buehler et al. [2001] compute the output image from any new viewpoint by taking the weighted average of reprojected images from the nearest input cameras. These nearby images are reprojected using some estimate of the geometry of the scene. For any image, the geometry is first textured by projecting the image on it. Then it is reprojected at the new position. When there are few input images, the computation of the geometry becomes very difficult and has many errors and undefined regions. The reprojected images do not align well due to these errors. When they are averaged, these appear as artifacts in the result (Fig:1.1a). Eisemann et al. [2008] try to remove these artifacts by aligning the reprojected images (Fig:1.1b). They use optical flow techniques to compute the shifts between these images. For any image, the flows to every other image are linearly combined to compute the shifted images which are then blended using weighted averaging. This works very well but requires computing optical flow for every pair of images. Though they mention that it could be possible to reduce "the quadratic effort to linear complexity by using intermediate results", optical flow techniques involve many computations and are not very fast. In their implementation, they blend only three reprojected images to reduce the computational cost.

In the context of image filtering, NL-means [Buades et al., 2005] denoises any image by replacing each pixel with the average of all pixels in the image which have similar neighborhood as the current one. The patchwise implementation replaces small patches of the image with the average of all patches similar to it. Eucledian distance is used to calculate the similarity between patches. Since this similarity does not look for patches which are well aligned, similar patches could have misaligned structures which are blurred after averaging. If the patches are aligned before they are averaged, the results could be sharper and closer to the desired image without any noise.

2.2 Phase-based Approaches

2.2.1 Background

We first provide a basic theoretical background of local phase before indroducing the approaches based on it. These concepts are explained in detail in Section 3.1. They are just introduced here for presenting the previous works.



(a) (b)

Figure 2.1: (a) A sinusoid, (b) its local phase computed using Hilbert transform.

Fourier Shift Theorem

The Fourier shift theorem relates the spatial shift of a signal with the shift in the phase of the Fourier transform. If the Fourier transform of a signal $f(x)$ is $F(\omega)$, the Fourier transform of $f(x)$ translated by $\delta(t)$, $f(x+\delta(t))$ is:

$$F^{'}(\omega) = e^{i\omega\delta(t)}F(\omega).$$

All the sinusoids with frequencies ω are shifted in phase by $\omega\delta(t)$. If we can compute these phase shifts, it is possible to manipulate them and thus, change the corresponding shifts in the spatial domain. For example, $f(x+\alpha\delta(t)) = F^{-1}(e^{i\alpha\omega\delta(t)}F(\omega))$ corresponds to the signal with $\alpha\delta(t)$ shift. Since the Fourier transform is global while the shifts on the images are local, the methods described later use local phase information instead of the global Fourier transform.

Local or instantaneous phase are the local properties of any signal and determines the position of any sample on a sinusoid like in the case of Fourier transform. The methods we use to determine the local phase localizes it in space and frequency. As such, for every frequency, we can shift any sample of the signal by manipulating its local phase. This comes directly from the Fourier shift theorem.

The most common way to compute the local phase of a 1-D signal is by using the Hilbert transform. More details about its computation will be provided in Section 3.1. Fig:2.1 shows the local phase for a sinusoid. The phase values are in the range $[-\pi,\pi]$ and the phase signal has many 2π phase-wraps. The local phase of a sinusoid specifies its position on it and thus, should be linear. The local phase computed in Fig:2.1b is linear but it has phase wraps whenever the phase goes beyond $[-\pi,\pi]$. It jumps to the phase value equivalent modulo 2π which lies in this range.

Complex Steerable Pyramid

We use a complex steerable pyramid [Simoncelli and Freeman, 1995] to compute the local phase of images. It decomposes any image into bands of different frequencies and orientations. The basis filters used resemble quadrature Gabor filters with fixed bandwidth. Quadrature Gabor filters have been used before [Fleet et al., 1991; Sanger, 1988; Fleet and Jepson, 1990; Gautama and Van Hulle, 2002] to compute the local phase in images. Each band (corresponding to a specific scale and orientation) of the pyramid is localized in space as well as the frequency.

Because of the complex-valued responses (by using even and odd valued filters like the quadrature Gabor filters), the local frequency and amplitude can be computed easily in each band of the pyramid. Wadhwa et al. [2013] provide a more detailed discussion on these pyramids as well as how they are relevant while using the properties of Fourier shift theorem.

Our main efforts while developing a method for blending will be to estimate the phase shift between the images well.

2.2.2 Disparity Computation

Disparity computation does not involve generating any new image from the inputs but use the phase information to estimate the shifts between the images. Local phase has been used to compute the horizontal disparity in stereo images [Fleet et al., 1991; Sanger, 1988]. Sanger [1988] computed the instantaneous phase of the images using quadrature Gabor filters. Their method is based on the Fourier shift theorem (Sec. 2.2.1). From the theorem, a phase shift of ωx is equivalent to a spatial shift of x if the frequency response of the filter is ω. They assign the phase difference between the two stereo images as the phase shift and use the center frequency of the Gabor filters as the frequency response. The disparity, then is calculated as $\frac{phase\ \ shift}{frequency\ \ response}$. Fleet et al. [1991] provided an improved algorithm where the disparity is defined to be the spatial shift required to make the phase signals of both images equal. If ϕ_l and ϕ_r are the local phase signals corresponding to the left and the right images, the disparity, $d(x)$ is the solution to

$$\phi_l(x - d(x)/2) = \phi_r(x + d(x)/2).$$

They iteratively solve for disparity using an optimization technique. They also determine regions of phase-singularity neighborhoods. These are the regions in the scale-space where the phase signal does not behave well. Specifically, when the complex response of the filters pass through the origin, the phase computation becomes unstable in the neighborhood. In their method, they discard the disparity values in these regions. They mention that it could be possible to use the information from neighboring points to estimate the information in these regions.

2.2.3 Optical Flow

Fleet and Jepson [1990] have used local phase information to compute the optical flow from a video sequence. They use a spatial-temporal filter in their method. They show that constant phase contours are stable information to track for computing the velocity. Gautama and Van Hulle [2002] provide another method for the optical flow estimation. They compute the local phase information using quadrature Gabor filters of constant bandwidth in many directions. They compute the velocity vectors in these directions by tracking contours of constant phases in time. They also determine the phase instabilities, which they call phase nonlinearity and remove the velocity components at the unstable points. For each velocity vector computed in a specific orientation, the full velocity is constrained to be on an infinite line orthogonal to the orientation of the filter. The full velocity is the point of intersection of all these constraint lines. They solve to determine the point closest to all of these lines.

2.2.4 Phase-based Image Blending

Wadhwa et al. [2013] manipulate the motion in a video using the local phase information. They use a complex steerable pyramid (see Sec 2.2.1) to compute the local phase at every frame. The phase information is band-passed across time to extract phases corresponding to a specific motion. This corresponds to the phase shift mentioned in Sec 2.2.1. They can then reconstruct a different signal by adding back a multiple of this band-pass which magnifies the motion by certain amount. They derive the bounds which identify the maximum magnification possible without many artifacts. They find out that the pyramids whose filters are more localized in frequency space would perform better and can magnify motion to a greater extent. This will be explained in detail in Section 4.3.

Didyk et al. [2013] use the same tools for view expansion for the application of automultiscopic 3D displays. They use stereo images and create neighboring views by extrapolating the local phases. The phase difference corresponding to the two images is considered as the phase shift, similar to Sanger [1988]. They manipulate this phase shift to generate new views. They also perform a regularization between levels of the phase difference. Because the phase values can only be between $[-\pi, \pi]$, phase shifts representing larger spatial shifts would be under-estimated. They identify the bands where phase shift is greater than $\pi/2$ and replace the phase shift in the band above as twice the shift in the current band. This step is important to represent large shifts between the images. This will be detailed in the next chapter where we will show that the phase differences may not be a good estimate for the shifts even if they are less than $\pi/2$.

A light field captures the spatial as well as angular information of the scene. It is a 4-D structure usually generated by an array of cameras or lenses. Zhang et al. [2015] reconstruct these light fields from micro-baseline image pairs. They avoid the issues in the phase-based disparity estimations by using the phase difference robustly, only to refine an initial disparity map. They compute the disparity between images using stereo matching. Since the input images are very close, the disparity may not be very accurate. They use the local phase information to refine and correct this disparity map iteratively. This works very well since phase measurements capture very small sub-pixel movements. Using the corrected disparity, they can reconstruct the 4-D light field using a disparity-assisted phase synthesis approach.

Local phase information for image interpolation has very recently been explored. Meyer et al. [2015] try to blend images in the context of video interpolation. They interpolate between the frames of any video using local phase. They regularize the phase information across scales using a different criteria and introduce a phase matching step which interpolates the motion of higher frequency better. The phase matching robustly removes the ambiguity because of the limited range of phase signals for small shifts.

— 3 —

Theoretical Background and Analysis of Synthetic Signals

We describe the theory of local phase and its computation from any input signal. We discuss the issues in representing shifts using phase differences and propose refinement techniques which can represent larger shifts than the current state of the art approaches. We develop the techniques by analyzing one dimensional signals. These are then applied to the two dimensional images to validate our observations.

3.1 Signals in one-dimension

As mentioned earlier, local phase along with the local amplitude describes the local properties of a signal. They are defined using the analytic signal, which is complex valued with only positive frequencies. In general, the analytic signal is defined as:

$$f_A(x) = f(x) + i f_H(x),$$

where $i = \sqrt{-1}$ and $f_H(x)$ is the Hilbert transform of $f(x)$. Hilbert transform is computed by shifting the fourier frequencies of the signal such that the new signal is analytic. Using $f_A(x)$, we can the calculate the local amplitude as [Granlund and Knutsson, 1995; Picinbono, 1997] :

$$A(x) = ||f_A(x|| = \sqrt{f(x)^2 + f_H(x)^2}.$$

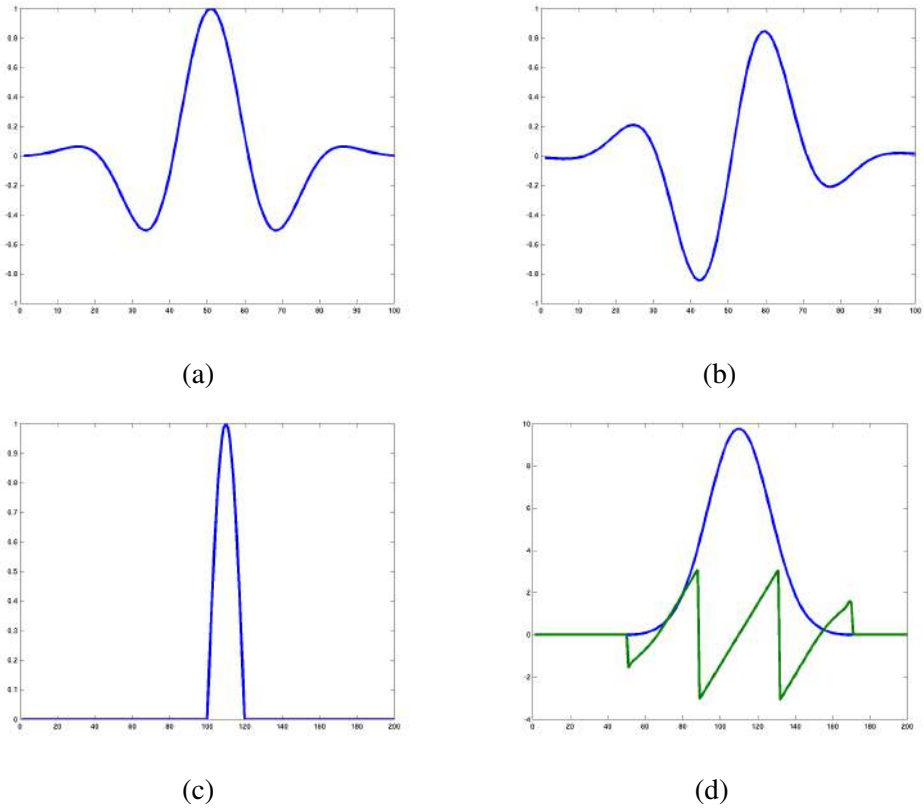
and the local phase as:

$$\phi(x) = arg(f_A(x)) = arctan(\frac{f_H(x)}{f(x)}).$$

From the analytic signal, the original signal can be reconstructed as:

$$f(x) = real(f_A(x)) = A(x)cos(\phi(x)).$$

Though the local phase as defined here is a local property of the signal, the computation of Hilbert transform uses the whole spatial and frequency domains. To analyze the signal better, it is important to determine local phases localized in the spatial and frequency domains. Computing the Hilbert transform of signals after bandpassing them localizes the phase in frequency while applying a spatial windowing function localizes it in space by restricting the computation



(a) (b)

(c) (d)

Figure 3.1: Quadrature-pair Gabor filters : (a) Real part of the filter; (b) Imaginary part of the filter (Hilbert transform of the real part); (c) A 1-D edge; (d) Its local amplitude and phase computed using these filters (blue - amplitude; green-phase).

to localized neighborhoods. The common way to do it is by using quadrature-pair filters. A quadrature-pair filter is the analytic representation of a real valued filter which can be used to compute local phase of the filtered function. Gabor filter is a good option since it is localized in frequency and space. The real part of the filter, which is the cosine or even symmetric part is represented as :

$$g_e(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} cos(2\pi\omega x),$$

where ω is the center frequency and σ defines the spread of the spatial window. Fig: 3.1 shows the real and imaginary parts of the Hilbert transform of this function. For any quadrature-pair Gabor filter, we can convolve it with the signal to get the local phase and local amplitude of the filtered signal(Fig:3.1d). These would now be localized within the band-pass of the filter in the frequency domain and the spatial window of the filter in the spatial domain.

.

3.1.1 Method

Consider two signals, where one is the shifted version of the other. We denote them as $f(x)$ and $f_{shift}(x)$. We compute the local phase of these signals, $\phi(x)$ and $\phi_{shift}(x)$ and the corresponding

local amplitude signals, $A(x)$ and $A_{shift}(x)$ by filtering the signal with quadrature pair Gabor filters. If $h(x)$ and $h_{shift}(x)$ are the response of $f(x)$ and $f_{shift}(x)$ respectively to the real part of the filter being used,

$$h(x) = real(A(x)e^{i\phi(x)}).$$

and

$$h_{shift}(x) = real(A_{shift}(x)e^{i\phi_{shift}(x)}).$$

We compute the phase shifts as the difference between their phases,

$$\phi_{diff}(x) = \phi(x) - \phi_{shift}(x).$$

The signal $h(x)$ with its phase shifted by half of $\phi_{diff}(x)$ can be calculated as:

$$h_{mid1}(x) = real(A(x)e^{i(\phi(x)-0.5\phi_{diff})}).$$

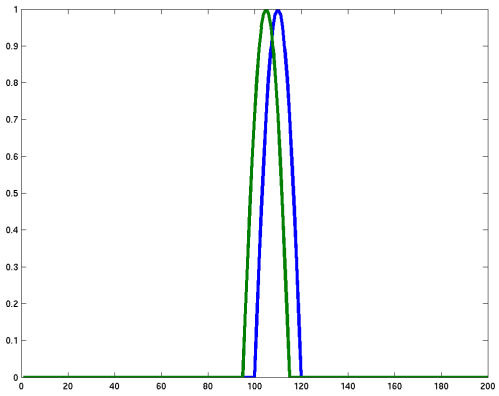
Similarily, $h_{shift}(x)$ shifted by the same phase towards the middle is:

$$h_{mid2}(x) = real(A_{shift}(x)e^{i(\phi_{shift}(x)+0.5\phi_{diff})}).$$

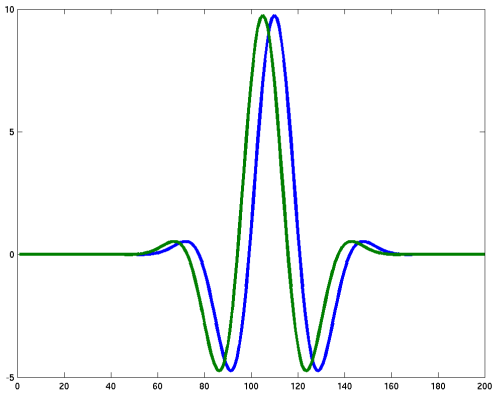
Shifting these phases would shift the signals in the spatial domain (fourier shift theorem). This would align the signals with each other. The final blended signal is computed by taking the mean of these two signals.

$$h_{blend}(x) = 0.5 * (h_{mid1}(x) + h_{mid2}(x)). \tag{3.1}$$

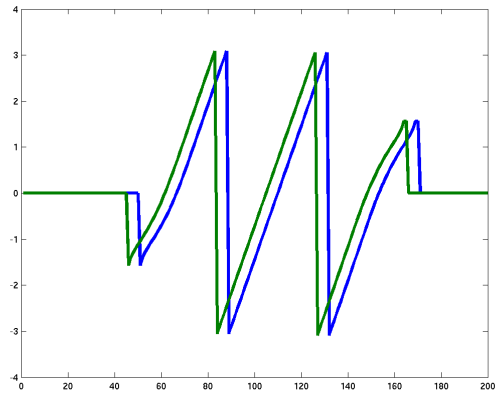
We only focus on blending the filtered signal here. Reconstructing the original signal from many filtered outputs will be discussed in the case of 2-D signals. It has been observed in previous work that the phase-based methods work well for low frequency parts of the signal but not for the high frequency parts. This was explained by Wadhwa et al. [2013] where they showed that the bound upto which the phases can be manipulated is inversely proportional to the frequency of the band-pass. The high frequency responses in the images often occur at the localized structures. We choose to analyze such structures. We use a synthetic 1-D edge and its shifted version as an example to explain the method. The edge can be represented as a half cycle of some sinusoid. Its local phase and amplitude are computed using the quadrature filters in Fig 3.1. Fig 3.2 shows the results for one value of shift. The blended result by doing this naively has many artifacts (Fig:3.2d). In the next section, we try to analyze and improve the method.



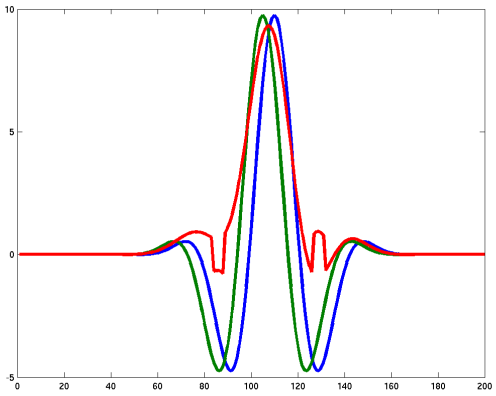
(a) Shifted signals



(b) Band-passed signals using Gabor filters



(c) Local phases of shifted band-passed signals



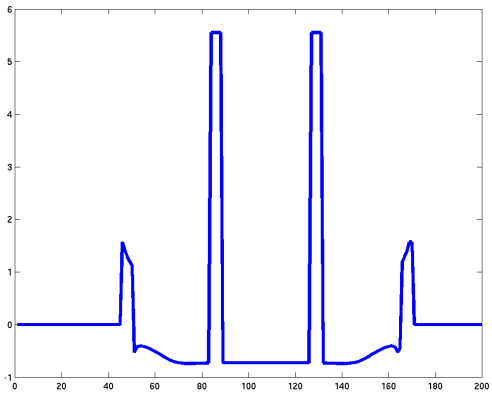
(d) Blended band-passed signal (in red)

Figure 3.2: 1-D example

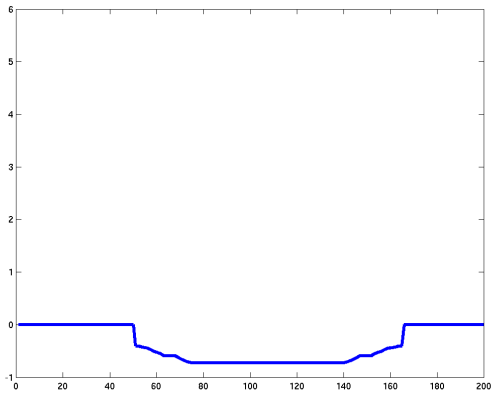
3.1.2 Refinement of phase difference

Median Filtering

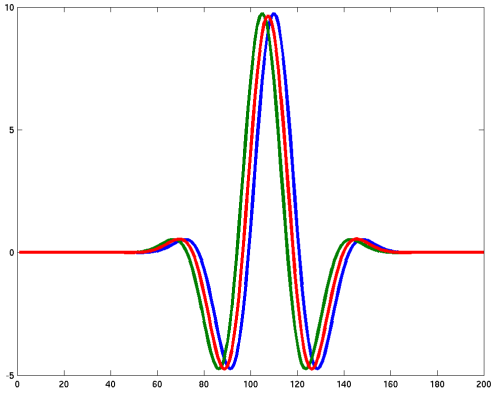
As explained in Section 2.2.1, the phase signal has many 2π phase wraps because of the limited range of the *arctan* function (Fig:3.1d). Computing the naive difference of phase signals thus, is not a good estimate for the phase shift between the signals. In Fig:3.3a , the peaks in the difference signal are because of the phase wraps. The phase differences around these peaks differ from the phase differences in the neighborhood by 2π. This shifts the signal around these peaks by different amounts and thus, the reconstructed blended signal might not resemble the shape of the shifted signals as seen in Fig:3.2d. The actual shift between the signals around these peaks is ambiguous by this factor of 2π. We remove this ambiguity and the artifacts created due to it by applying a median filter to the phase difference before using it to compute the blended signal. The median filter removes these peaks and makes the phase difference consistent, substantially improving the result (Fig:3.3). The size of the median filter is fixed to the period of the sinusoid being used in the Gabor filter. This removes all the peaks from the difference signal. Note that this step might not choose the correct phase difference among the ambiguous options (the peak could have been the correct difference) but the phase difference



(a) Phase difference



(b) Median filtered phase difference



(c) Blended signal (in red) using the median filtered phase difference

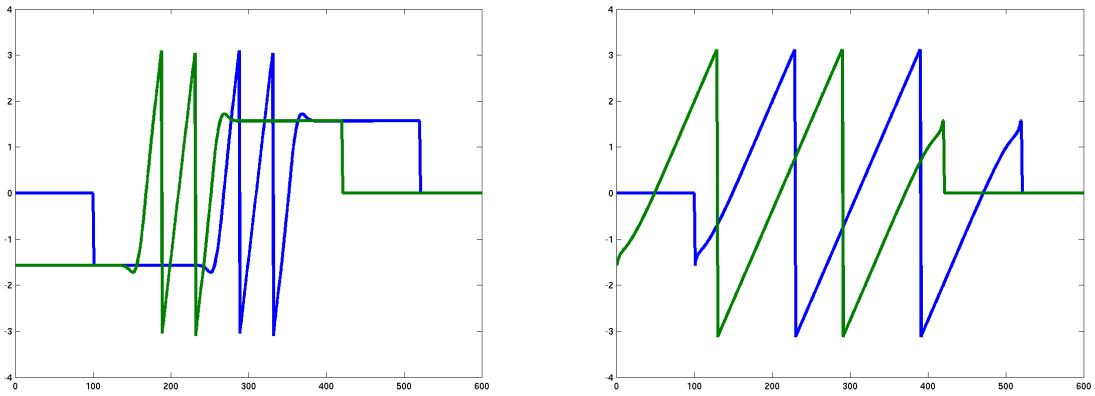
Figure 3.3: Median Filtering

values will be consistent and will not distort the shape of the signal. We discuss about correcting for the incorrect phase estimates of phase difference in the next section. After the peaks are removed, the resulting phase differences would lie in the range $[-\pi, \pi]$. In Fig:3.3a, the first and the last peak are due to no overlap of the corresponding phase signals (Fig:3.2c). The median filter also removes these artifacts from the phase difference.

Regularization across different scales

[Didyk et al., 2013] replaced the phase difference in the higher level whenever the phase difference in the lower level was greater than $\frac{\pi}{2}$. This was because the phase difference would be under estimated for the large shifts. We show that this regularization is important even when the phase difference in the lower level is smaller than $\frac{\pi}{2}$ by studying the issues with representing shifts with the phase difference.

As observed before, there is some ambiguity in determining the correct phase shift from the phase difference. In fact, it is not the only source of ambiguity. After median filtering, the phase differences are restricted to the range $[-\pi, \pi]$. The phase difference of 2π represents a



(a) (b)

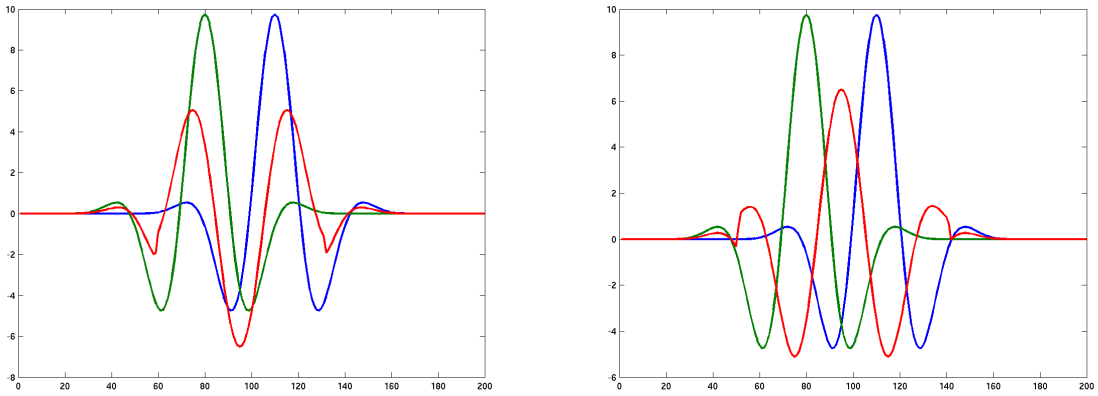
Figure 3.4: Local phase of two signals with a large shift: (a) difference computed using filter with high center frequency; (b) computed using filter with 0.25 times the center frequency of filter in (a). Here, the local phases in the higher frequency do not overlap each other at all while in the lower frequency, there is sufficient overlap.

shift equivalent to the period of the sinusoid currently being used to compute the phases. Since the lower frequencies have higher periods, large shifts are better represented in the lowest levels. If the shift is higher than what can be represented in the phase difference, it is estimated incorrectly by some multiple of 2π. In practical cases when the signals are far, the phase difference might not be a good estimate for the phase shift even by some factor of 2π. This is when the structures are local. In this case, because of the limited spatial support of local phase, the phase difference would not be capturing the phase shifts between the same structure and thus, would be completely wrong (Fig:3.4a). To have some overlap between the phase signals corresponding to some shifted structure, the shift between the structures should be within the width of Gabor filter being used. The phase signals would have more overlap in the lower frequencies since the width of these filters would be larger (Fig:3.4b). Since the estimates are better in the lower frequencies, we can use the correctly estimated phase shifts in the lower levels to correct for the phase shifts in the higher level.

We assume that the shifts in the lowest level (corresponding to the lowest frequency Gabor filter) can be represented by the phase difference. This is reasonable since the phase shifts in the lowest level should be very small. From the Fourier shift theorem, we know that the phase shift between samples which have a spatial shift of x units is ωx, where ω is the frequency in the Fourier decomposition. In our case, ω is the center frequency of the band-pass. It is possible to use the phase shifts in the lower frequency to correct the phase shifts in the higher frequency. Suppose we have two quadrature pair filters with center frequencies ω_{high} and ω_{low} with $\omega_{high} > \omega_{low}$. The phase differences in the higher level, $\phi_{diff_{high}}$ and in the lower level, $\phi_{diff_{low}}$ should ideally be related by :

$$\phi_{diff_{high}} = \frac{\omega_{high}}{\omega_{low}} \phi_{diff_{low}}. \tag{3.2}$$

We can replace the phase shift in the higher frequency by the modulated phase shift in the lower frequency if we know that the higher frequency estimates can be inaccurate. From Fig:3.5, we



(a) (b)

Figure 3.5: 1-D Regularization : (a) The blended high frequency signal(in red) using the estimate of phase shift from the filter with the same frequency; (b) The blended high frequency signal using the estimate from the filter with center frequency 0.5 times the higher one.

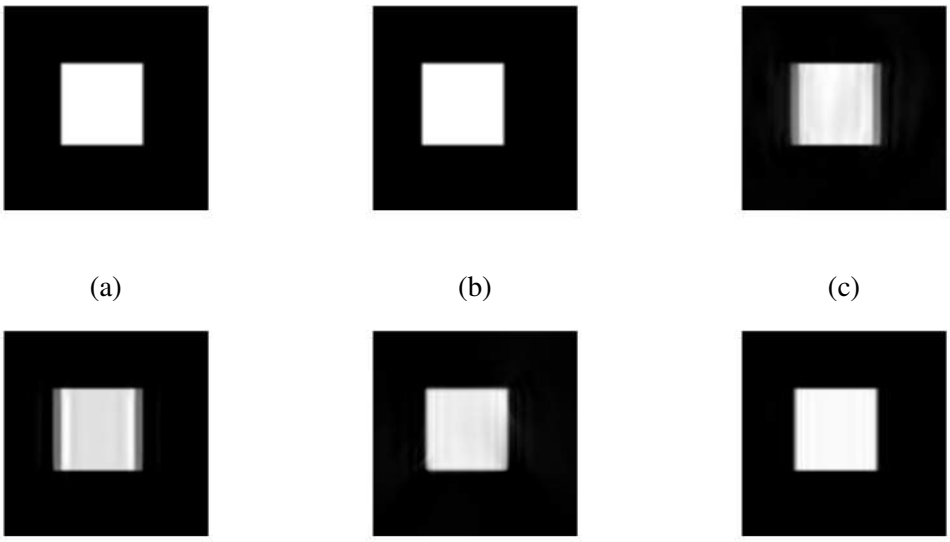
can see that the blended signal is estimated better after this regularization. Since we don't manipulate the amplitude signals, the signals are not estimated exactly as seen from the attenuation of peaks in the blended signal (Fig:3.5b).

To summarize, the method first refines the phase shift using median filtering and cross-scale regularization of the phase difference between the signals. Then, it blends the signals according to equation 3.1 where $h_{mid1}(x)$ and $h_{mid2}(x)$ are computed using the refined phase shift.

3.2 Signals in two dimensions

In the case of images, we use the output of a complex steerable pyramid [Simoncelli and Freeman, 1995] which decomposes the image at many scales and orientations. The pyramid also stores non-oriented high-pass and low-pass residuals which are not captured by the band-passed levels. The basis filters used resemble the quadrature-pair filters. For each level and orientation, the Hilbert transform is computed along that orientation to get the analytic signal. The pyramid is very useful for phase manipulations or manipulating shifts independently in any band since it is translation invariant. We can compute the local phase and amplitude of the image at any scale and orientation using the same approach as the 1-D case since we have an analytic function for every band. The pyramid is self-inverting and uses the same transfer functions to reconstruct the image from its decomposition. There are other approaches which can compute isotropic analytic signals at multiple scales by using high dimensional Hilbert transform for images [**?**] but we only consider the steerable pyramid in our work.

Similar to 1-D, we compute the phase difference and use it to shift the phases of the two images. We refine the phase difference using median filtering and cross scale regularization. The filters used to build the pyramid have equal log width in the frequency domain. We use the center frequencies of the corresponding bandpass filters to regularize these bands. In the case of the complex steerable pyramid, the center frequency of the filter corresponding to the higher



(a) (b) (c)

(d) (e) (f)

Figure 3.6: 2-D blending : (a),(b) Images shifted by 10 pixels; (c) Blended image without refining phase difference; (d) Using median filtered phase difference but no regularization; (e) Using regularization but no median filtering; (f) Using regularization and median filtering.

frequency is twice the center frequency of the filter corresponding to the lower frequency for consecutive bands,

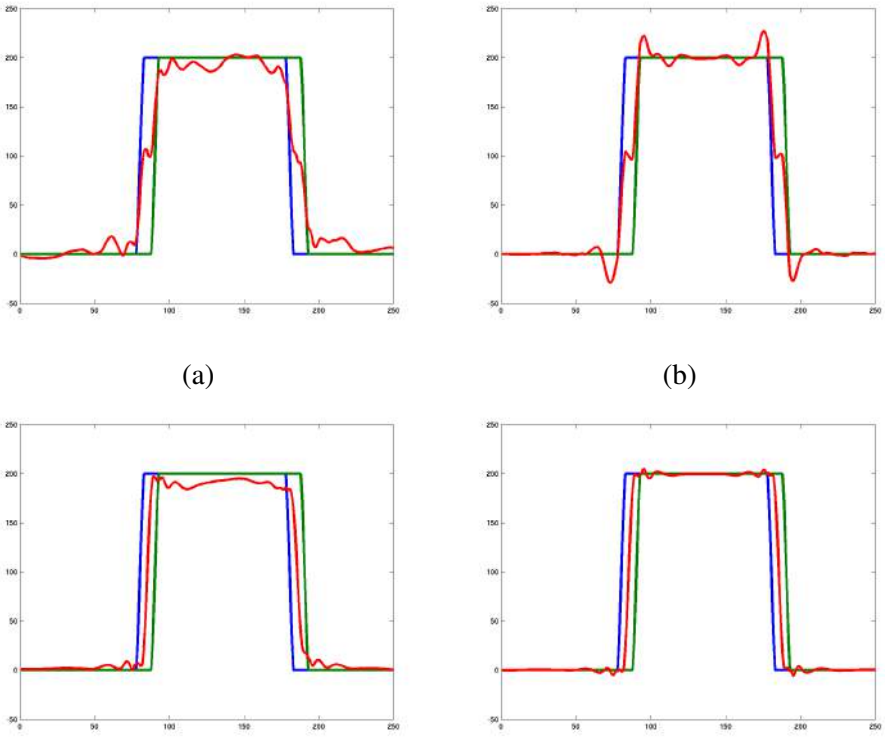
$$\omega_{high} = 2\omega_{low}$$

and from equation 3.2,

$$\phi_{diff_{high}} = 2\phi_{diff_{low}}$$

Like in the case of the 1-D example, we replace the phase shift in the higher level to double of the phase shift in the lower level. The final blended image is the average of the two phase-shifted images. We don't manipulate the high and low pass residuals of the pyramid. If the pyramid is computed using many levels, these residuals should not have a lot of information about the structures in the image.

We consider a square in the image which is shifted horizontally by 10 pixels in the second image. The local phase is computed using a complex steerable pyramid with five scale levels and five orientations. Fig:3.6 illustrates the blended signal by using different phase refinement techniques discussed. The results support our analysis. The regularization helps in recovering the phase shifts in regions of no overlap but does not necessarily preserve the shape of the signal. Using median filtering and regularization together gives a good estimate of the image in between. In Fig:3.7d the ringing around the edges happens due to the regularization. Since the spatial support of the filter in the lower level is larger, neighboring pixels are affected in the higher frequency too. This results in these pixels being shifted by some amount. We also lose all phase shift information which was exclusively in the higher frequency. In the case of structures, all frequency components should be present and thus, structures like edges are shifted well but there is ringing. Fig:3.7 demonstrates the quality of our result by showing a horizontal scanline corresponding to the square in the images.



(a) (b)

(c) (d)

Figure 3.7: A scanline of the images. The shifted input signals are in blue and green. Blended signal (in red) computed (a) without any refinement of phase difference; (b) with median filtering but no regularization; (c) with regularization but no median filtering; (d) after median filtering and cross-scale regularization of phase difference.

— 4 —
Experiments on Real Images

Based on the method developed, we will align real shifted images. We will refine the method by introducing a local regularization which aligns the images better. We will also develop a method to interpolate between the images and to align four images.

4.1 Two images

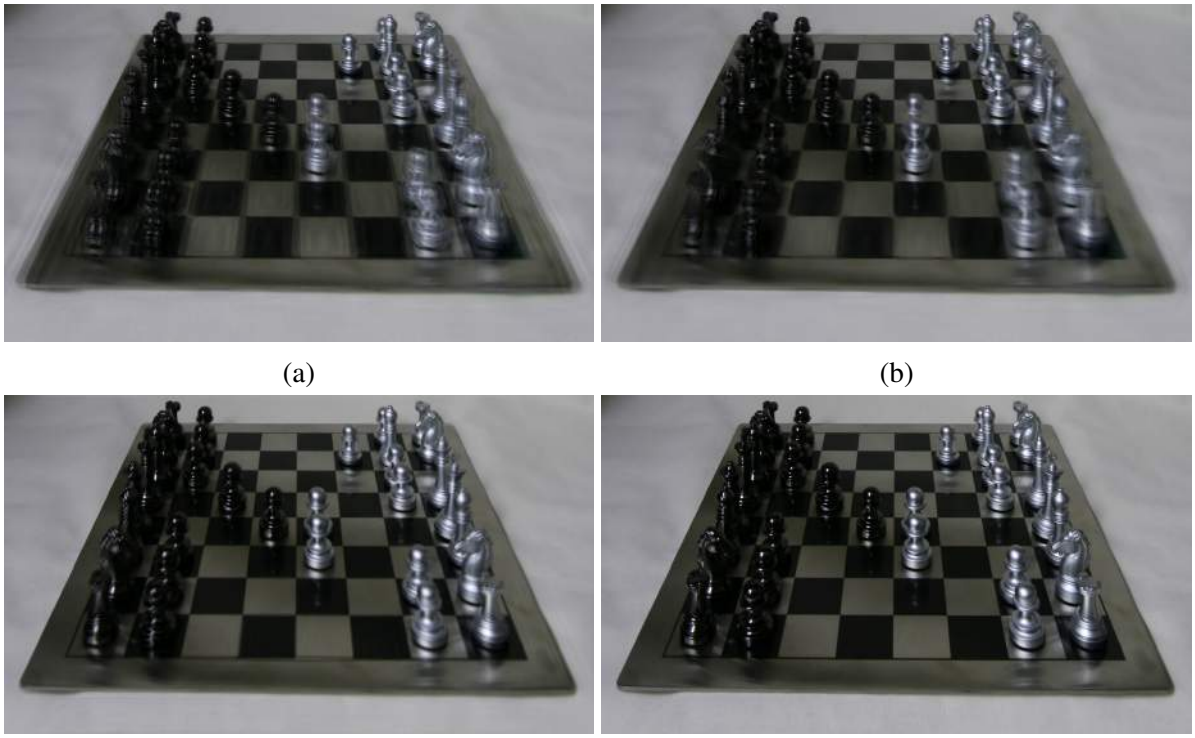
Here, we consider two images from a lightfield. We compute the image in the middle of these images. The results are shown in Fig 4.2 for input images with a maximum of 20 pixels shift. The ground truth for the image in the middle can also be obtained from the light-field (Fig:4.2d). With no phase refinement, the blended result is not aligned as well as very noisy. This is as expected since the images are far and the phase signals do not overlap in the higher frequencies. When the phase difference is regularized over scale, the blended image is aligned well (Fig:4.2b) but the result is very distorted. Note that the artifacts increase with increasing shifts. The shifts in the back are less and these areas thus, have less artifacts than other areas which have larger shifts. To preserve the shape of the output, we median filter the phase differences before regularizing them over scale. The result is much cleaner and its quality is close to the ground-truth (Fig:4.2c).

4.1.1 Local Regularization

Though the cross scale regularization corrects the incorrect estimates of phase difference at the higher levels, we can only estimate the shifts corresponding to the lower bandpass information

(a) (b) (c)

Figure 4.1: (a) and (b) are the shifted input images. (c) is the naive blending (mean of the images).



(a) (b)

(c) (d)

Figure 4.2: Blended Image: (a) Without any phase refinement, (b) with regularization but no median filtering, (c) with regularization and median filtering. (d) Ground Truth.

after the regularization. We lose all the phase shift information which was present in the higher level but not in the lower level. In this case, the regularization would underestimate the phase shifts in the higher frequency. Instead of replacing all the pixels in the higher frequency band of the phase difference with the lower frequency band, we make the choice of regularization independent for all the pixels. Local regularization preserves the higher frequency content with small shifts which can be estimated well. For each pixel in the lower level, if the phase difference is higher than the threshold, we replace the corresponding pixel in the higher level with the manipulated phase difference at the lower level. This ensures that different parts of the image are regularized from different levels and avoids some parts of the image being over-regularized. It reduces the extent of underestimation of phase shifts. In Fig:4.3a , the shifted images are not very well aligned and the blended image has double edges. This is because the phase shifts are under-estimated in some regions (Fig:4.3c). Local regularization (Fig:4.3b) aligns the images well since the phase shifts are not as underestimated as before. All the subsequent results will be demonstrated using this local regularization.

Interpolation

In addition to computing an image in the middle of the inputs, we can interpolate between them too. The equation 3.1 is modified to

$$h_{blend}(x) = y * h_{mid1}(x) + (1-y)h_{mid2},$$

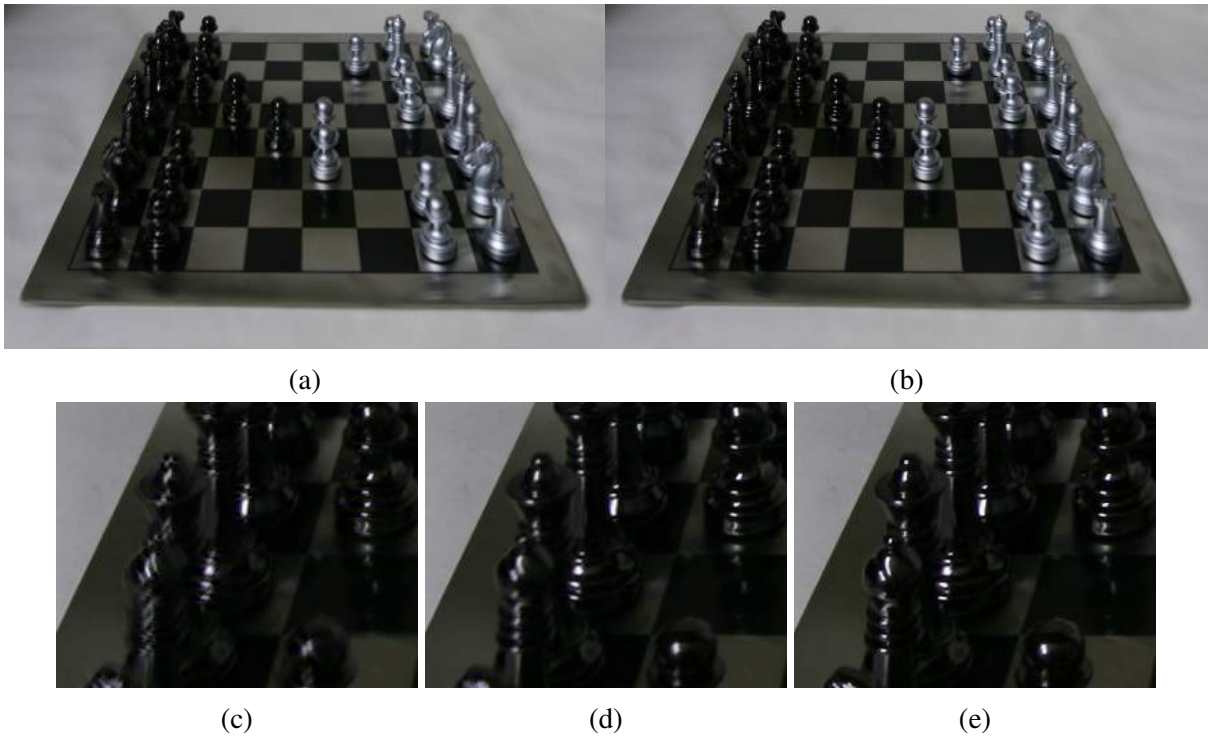
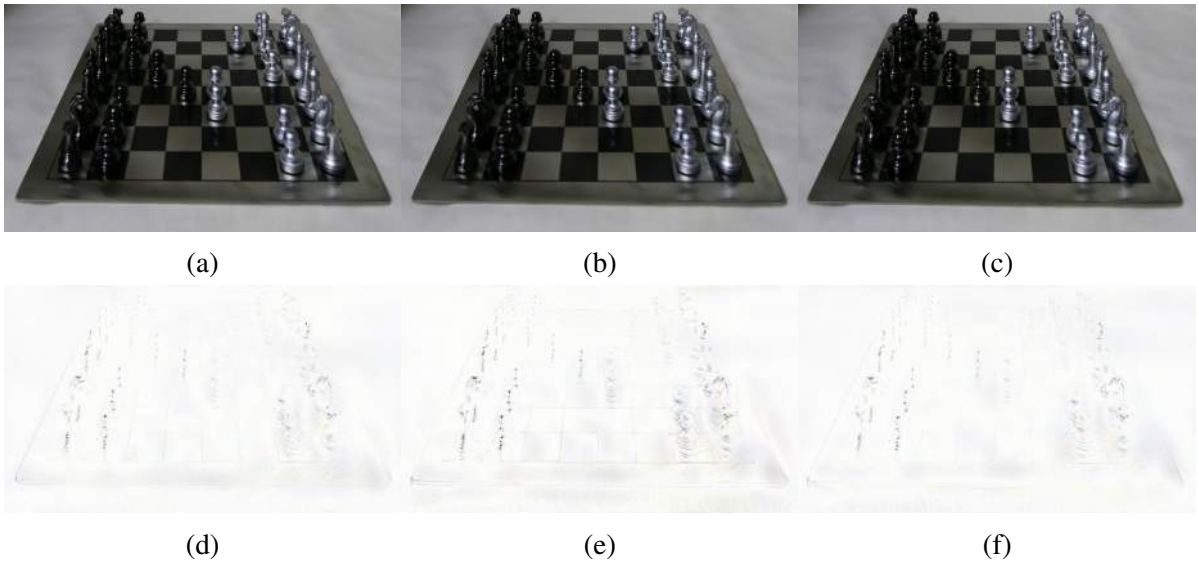


Figure 4.3: Local Regularization : Blending with (a) global regularization; (b) local regularization. Second row shows a cropped region from these images for (c) global regularization; (d) local regularization; (e) ground truth.

where $y \in [0,1]$. Here, $h_{mid1}(x)$ is calculated by shifting the phase of $h_1(x)$ by $y\phi_{diff}$ and $h_{mid2}(x)$ by shifting the phase of $h_2(x)$ by $(1-y)\phi_{diff}$. This ensures a smooth transition between the interpolated images. The image closer to the position is given more weight since there are more artifacts when the images are shifted by a large amount. Fig:4.4 demonstrates this. The interpolated images are well aligned with the ground truth but the difference is maximum in the middle and decreases as we go towards any of the input images. This is because in the middle, both input images have to be shifted by a large shift while as we go towards one of the inputs, one of the images is shifted by a smaller amount. The image shifted by a smaller amount is estimated better and is given more weight while computing the blended image. The sequence of interpolated images transition smoothly without any visual incoherent jumps for different regions of the image.

Fig:4.5 shows the blended image in the middle for inputs with increasing shifts. As the shifts increase, the results degrage. When the shifts are very small (Fig:4.5b), the blended image is very well aligned and sharp (Fig:4.5a). The overhead cables in the image are thin structures with low spatial extent and high spatial frequency. These would not be present in the low bandpassed levels. Regularization in that region would underestimate the shifts for the wires. As the shifts increase, these wires are not aligned properly. In Fig:4.5e, they are not aligned at all and thus, the blended image has ghosting artifacts similar to linear blending. Trees have high frequency content which cannot be represented by the lower levels well. If the tree shifts by a large amount, the higher frequency phase difference will be replaced by the lower frequency contents. This blurs the output as seen from the images. The building and other structures are preserved and shifted well for larger shifts as they have information in all



(a) (b) (c)

(d) (e) (f)

Figure 4.4: The first row shows three interpolated images between two shifted input images. The second row shows the complement of twice the difference of these computed images from the ground truth. Darker color signifies larger difference.

frequency bands which are coherent. As the shifts increase, larger structures start to produce artifacts. In Fig:4.5e, the pole infront of the tree is not estimated well. This is because the level at which the regularization started, the pole was not represented well which resulted in the phase shifts being underestimated at the higher levels.

4.2 Four Images

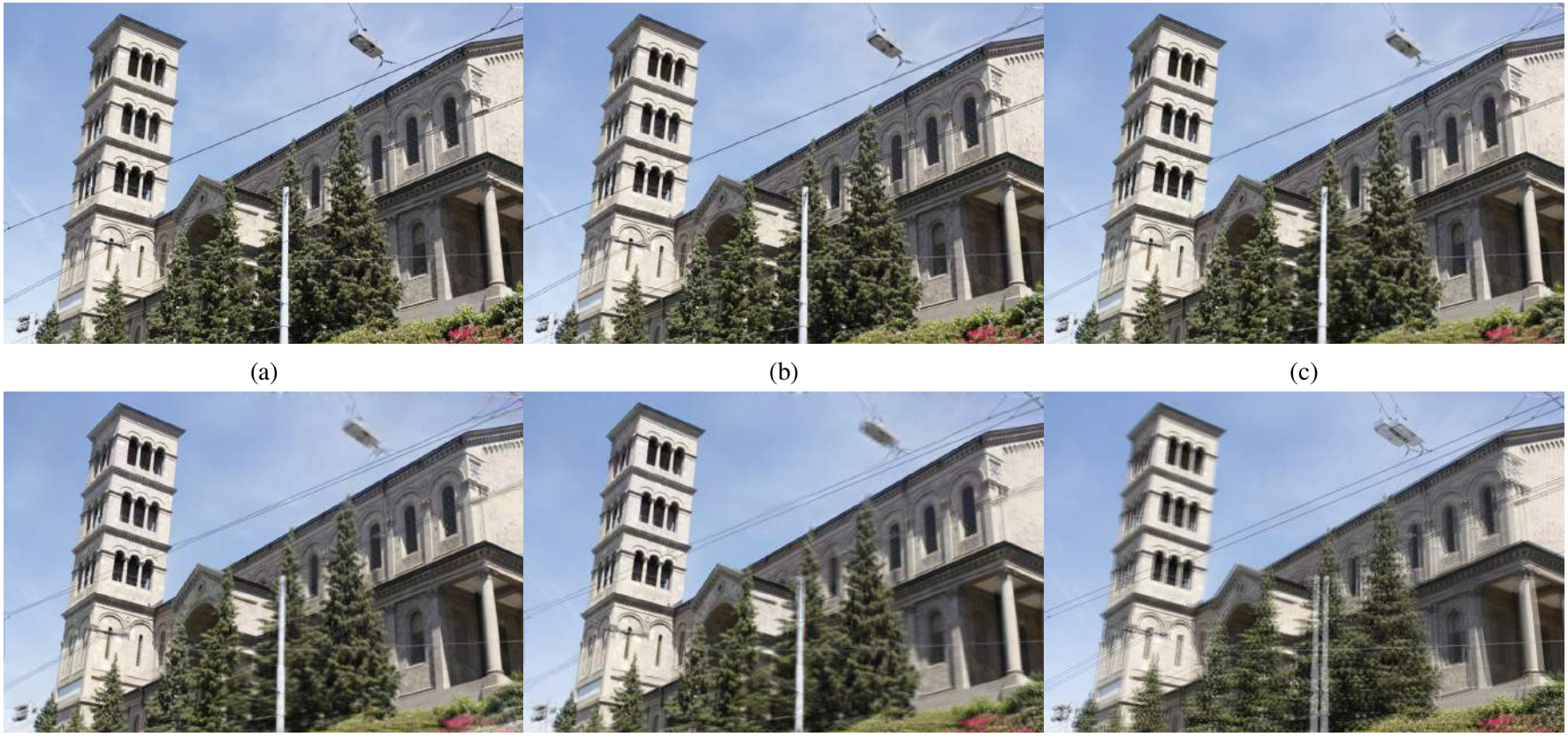
We extend our approach to blend four images. In this case, we assume that the image in the middle should have the mean phase of these images. On this assumption, we shift the phase of each image to the mean phase. If the amplitude and local phase decomposition of image i = $\{A_i, \phi_i\}$, $i \in [0,3]$,

$$\phi_{mean} = \frac{(\phi_0 + \phi_1 + \phi_2 + \phi_3)}{4}$$

$$\phi_{diff_i} = \phi_i - \phi_{mean}$$

$$\phi_{shift_i} = \phi_i - 0.5\phi_{diffi}$$

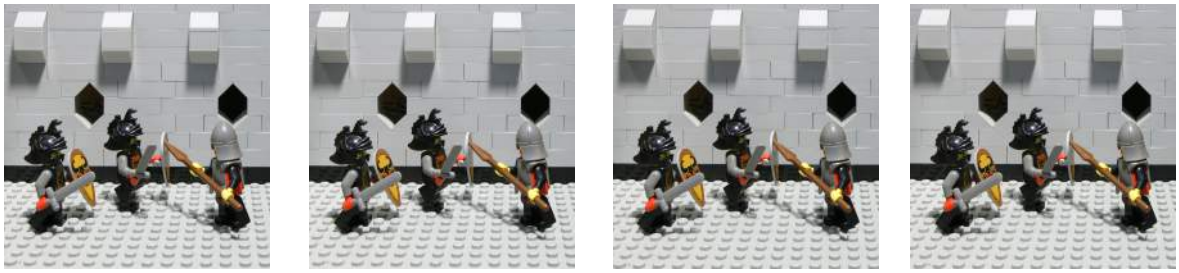
The final output is the mean of the four images whose local phases have been shifted towards the mean phase i.e. the images have been reconstructed using $\{A_i, \phi_{shift_i}\}$. As done before, the phase difference is refined before regularizing. Fig:4.6 and 4.7 demonstrate blending four shifted images with maximum shift of around 20 pixels. Using the formulation for four images, it is easy to extend the approach to any number of input images. We show results only for four images here.



(a) (b) (c)

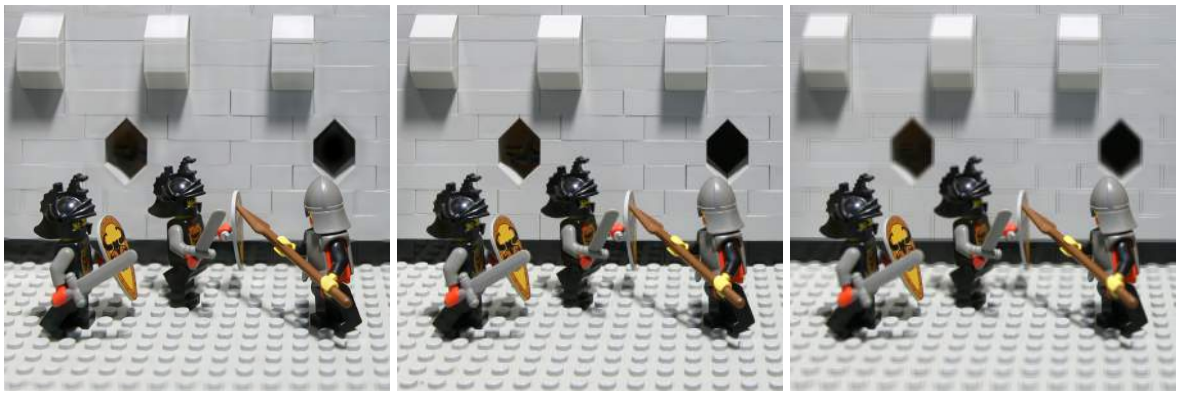
(d) (e) (f)

Figure 4.5: Blending two images : The blended image in the middle degrades as the shifts between the inputs increase. (a) One of the input images. (b) - (e) Blended image with increasing shifts between the inputs. The increase in shifts is not linear. Maximum shift between images in (e) is 25 pixels. (f) shows the naive blending between the input images in (e). Images taken from Kim et al. [2013].



(a) (b) (c) (d)

Figure 4.6: Shifted Input Images



(a) (b) (c)

Figure 4.7: Blending 4 images: (a) Output after phase refinement, (b) Ground Truth, (c) Naive blending (mean of the input images).

4.3 Sub-octave pyramids

From our analysis of phase difference, it is clear that larger spatial support of filters encode large shifts better as they ensure more overlap of phase signals. In addition, [Wadhwa et al., 2013] showed that there is a limit beyond which shifting the local phase can introduce artifacts even with a very good estimate of the shift. This is also because of the limited spatial support of the filters used to reconstruct the image. They derive a bound beyond which the phase should not be shifted. The original pyramid has approximately one period of the sinusoid under a Gaussian window of the Gabor filter. They propose using a sub-octave pyramid which has more periods of the sinusoid under the Gaussian window. This would allow greater phase shifts since the filters now have more spatial width.

We use a half-octave pyramid which has two periods of sinusoids under the Gaussian window at any frequency of the pyramid. The regularization works similarly as described in equation 3.2. The half-octave pyramid would have lesser width in the frequency domain as it is wider in the spatial domain. For the half-octave pyramid, $\omega_{high} = \sqrt{2}\omega_{low}$. The rest of the approach remains the same.

In Fig:4.8c, there are some ringing artifacts around the edges which are removed when half-octave pyramids are used (Fig:4.8d).

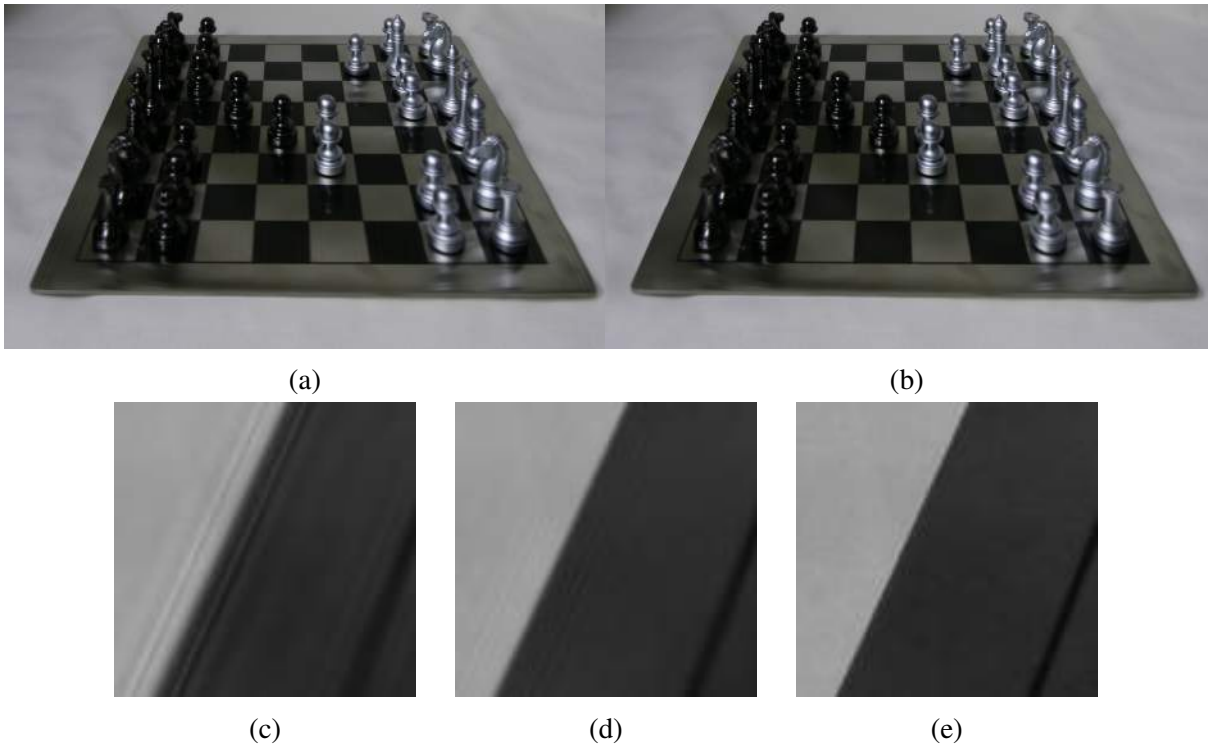


Figure 4.8: Blending two images: (a) Output with octave pyramid, (b) Output with half-octave pyramid. (c) and (d) show a cropped region from (a) and (b). (e) shows the ground truth for this region.

4.4 Comparison with other alternative approaches

Two current approaches for blending are naive mean of the images or using the optical flow estimation to blend the images.

Optical Flow has been used for blending by Eisemann et al. [2008]. Optical flow between all pairs of images are computed. If $W_{I_i \to I_j}$ is the flow from image I_i to image I_j, the flow for warping image i is

$$W_{I_i} = \sum_{j=1}^{n} \omega_j W_{I_i \to I_j}$$

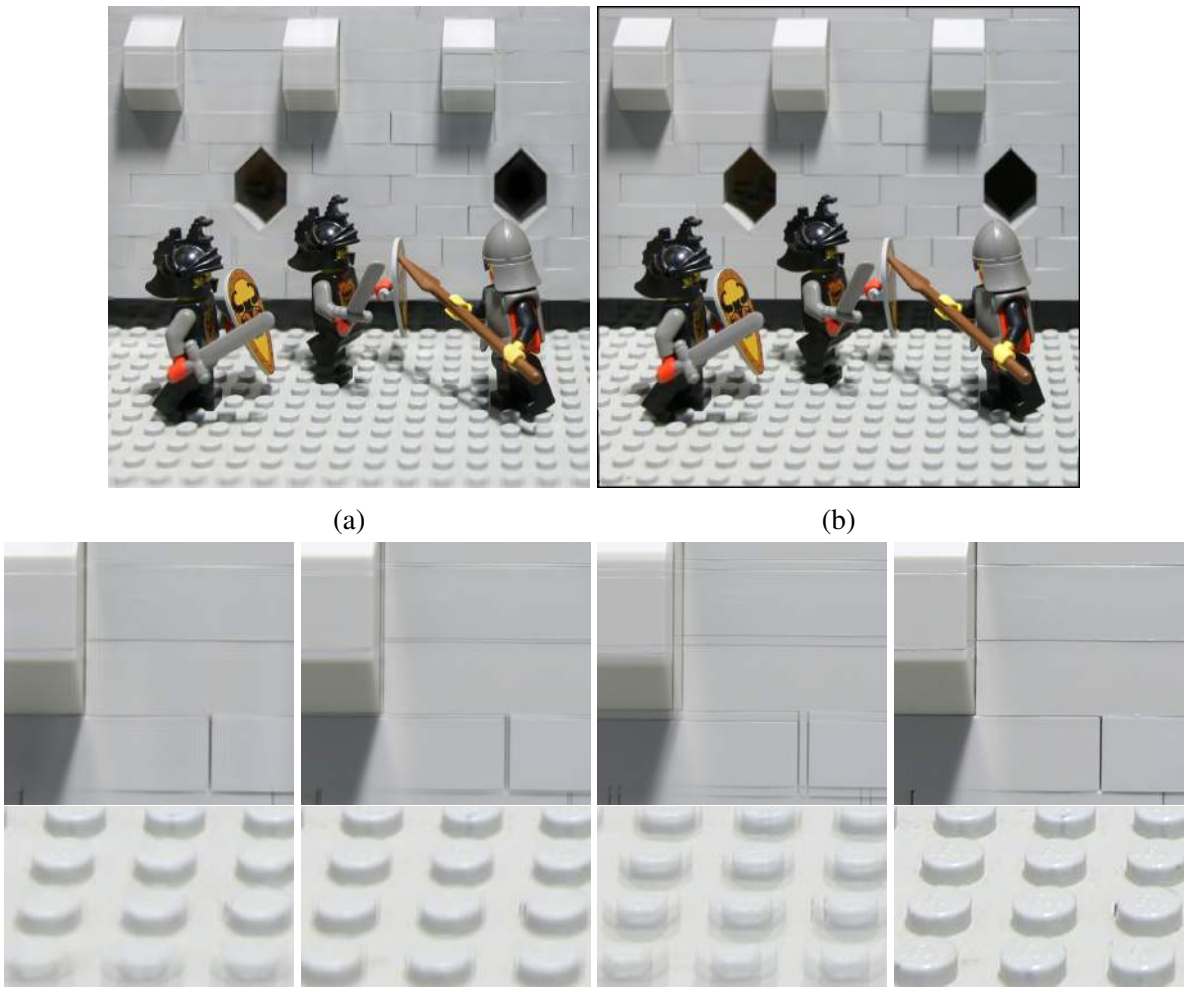
where ω_j are the weights assigned to each warp. They claim that their results are independent of these weights as long as their sum is 1. We choose $\omega_j = \frac{1}{3}, \forall j$ for blending four images. We use the same optical flow method as used by them [Brox et al., 2004].

Our method (Fig:4.9a) is better than the naive approach (Fig:4.7c) in all the cases since it aligns the images better. Fig:4.9 compares our method with optical flow method of Eisemann et al. [2008] for blending four images. When the shifts are small, out method gives similar output as optical flow. When the shifts increase, optical flow gives a sharper and better aligned output. In Fig: 4.9 for the cropped regions in the first row, our method gives a sharper result. This shows that phase difference can accurately represent the shifts if they are small. For the region in the second row, optical flow gives a sharper output. The shifts in this region are larger than the region in the first row. The results are sharper because of the optical flow computation as well as the fact that pixels are moved spatially after the computation of flows. In our case

when the shifts become large, shifting phase can itself introduce artifacts.

In terms of speed, the optical flow blending takes around 500 seconds to compute the blended image from four input images of size 800*1400 while the phase based approach takes around 50 seconds with global regularization and 140 seconds with local regularization using the half-octave pyramid of 14 levels and 6 orientations inn MATLAB. In general, the phase-based method is 3-4x faster than the optical flow approach. Eisemann et al. [2008] use a GPU implementation of optical flow and achieve a speed of 5 to 24 fps at 1027*768-pixel rendering resolution for the IBR system. All the techniques used in our approach are highly paralellizable. The construction and reconstruction of pyramids only involve convolutions, which is easily paralellizable. The median filter can be independently applied to all bands. Cross-scale regularization is not paralellizable within any orientation but the computation can be independently done for all orientations. Meyer et al. [2015] implemented their phase-based interpolation method on GPU and reported a substantial speedup over the optical flow method of Brox et al. [2004]. The amount of computation required in our approach only depends on the size of the images and the shifts between them. It scales easily with the size of the images as well as the number of images.

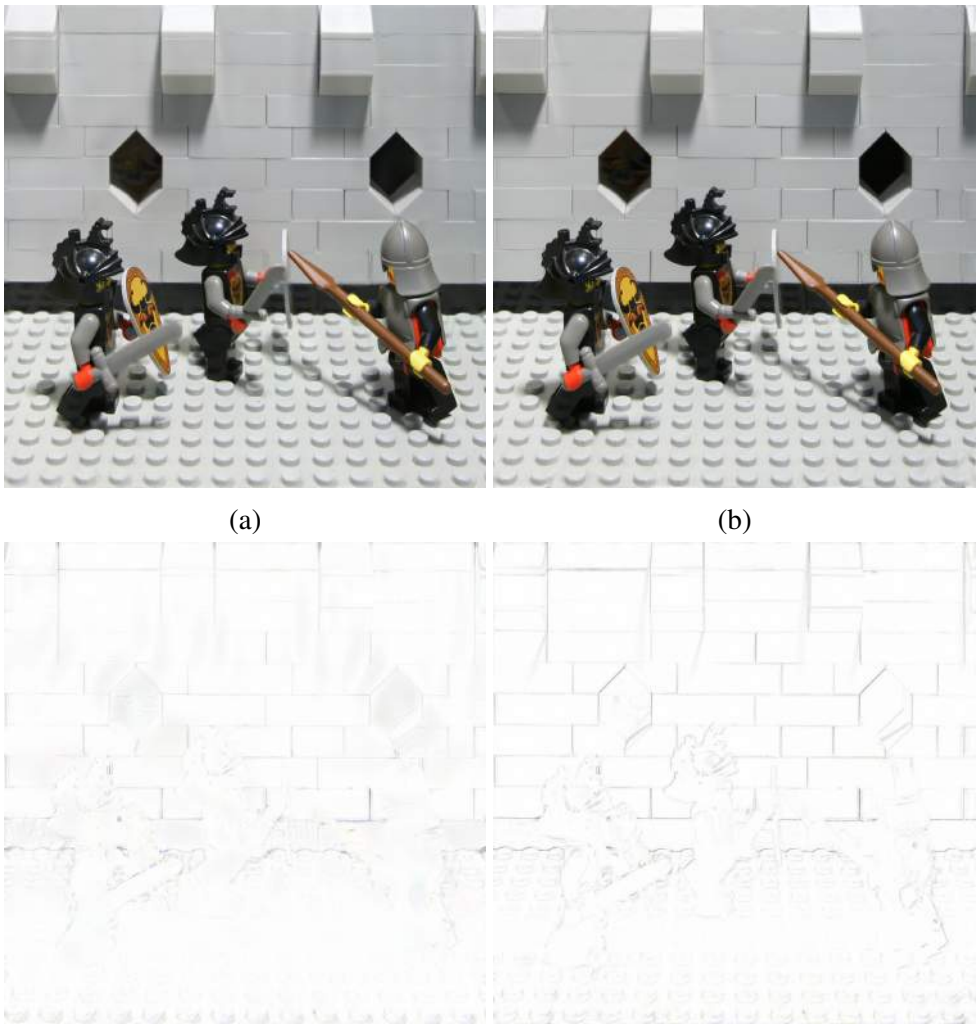
For phase-based methods, [Didyk et al., 2013] extrapolated a pair of images using only regularization as the phase refinement. We compare to this method throughout while explaining our solution. Our method is significantly better than their approach. Very recently in parallel to our work, Meyer et al. [2015] have tried to interpolate frames of video and have much better results than the method of Didyk et al. [2013]. They incorporate a phase-matching step after a cross-scale regularization similar to ours. They try to manipulate the original high frequency phase differences by multiples of 2π to be as close as possible to the regularized phase difference. This preserves the details in the high frequency while resolving the ambiguity upto some factor of 2π. Fig:4.10 compares our results to them. The quality of the output is similar in their case but our output aligns better with the ground truth (Fig:4.10c) than theirs (Fig:4.10d). This is because we compute the output by blending the two shifted input images while they shift only one of the images to compute the output. Their output is sharper for information which is only present in some high frequency bands (Fig:4.11). While regularizing, we would lose the phase shift information for those regions. They manipulate the original phase differences to be as close as possible to the regularized phase. In this way, they can retain some high frequency information.



(a) (b)

(c) (d) (e) (f)

Figure 4.9: Blending four images: (a) Output using our method, (b) using optical flow. The next two rows show cropped regions from images. (c) are regions from output using our method, (d) using optical flow, (e) using naive blending (mean). (f) shows the same regions from the ground truth. For the regions in the first row, our method performs better while for the second row, optical flow result is sharper.



(a) (b)

(c) (d)

Figure 4.10: Comparison between our method and the output computed by Meyer et al. [2015]. (a) is the output of blending two shifted images using our method, (b) output image using their method for the same input, (c) complement of the difference between our output and the ground truth, (d) complement of the difference between their output and ground truth. Darker color signifies larger difference.

(a) (b)

Figure 4.11: Comparison between our method and the output computed by Meyer et al. [2015]. (b) Their method gives a sharper output than (a) ours for regions with information only in the high frequencies. Image taken from [Meyer et al., 2015].

— 5 —
Possible Applications

We have developed a method to align images with small shifts between them. It is possible to use this method to improve the applications mentioned in Chapter 1. We discuss the details here and describe the issues in using phase-based approaches for these applications.

5.1 Image Based Rendering

As explained before, image based rendering methods blend reprojected images captured by the nearby cameras to compute the image from any new camera position. The usual way to blend these images is to take a weighted mean. If these image are not aligned, the output has ghosting artifacts. We can use our approach to blend the images instead of the linear blending. This would require computing the complex steerable pyramids for these images during the rendering. The local phase can be computed from these pyramids and can be used to blend them.

As mentioned while comparing our algorithm to other approaches, the pyramid computation is highly parallelizable and the implementation of Meyer et al. [2015] is much faster than the optical flow techniques. This could make it easy to blend more than 3 images in the case of Eisemann et al. [2008].

Reprojecting the input images involves warping them according to the geometry of the scene and the camera positions. It may be possible to pre-compute the pyramids for all the input images and warp the corresponding local phase and amplitude information similar to warping the original images. This would be very fast since there would not be any pyramid construction during the runtime. The method would only require reconstructing the blended image using the new pyramid created by our method. This would involve some overhead with respect to storage.

The steerable pyramids are over-complete. Because of multiple orientations at different levels, they occupy a lot of space in the memory. If there are h levels and o orientations, the size of the steerable pyramid computed for an image whose size is M pixels would be $M(h*o+2)$ pixels. The $2M$ pixels are for the high-pass and the low-pass residuals. This size is reduced by using sampling theory. Since the information at the lower level is at a lower frequency, it can be represented with smaller number of samples. In the case of the octave pyramid where the center frequency of the higher level is twice the center frequency of the lower level, the lower level can be represented as half as many samples as required to represent the higher level. This also reduces the speed of construction and reconstruction of these pyramids since

the number of convolutions with the basis filters is reduced. All the results shown in this work use this representation of the pyramid. The pyramid now has at max $M * (2o + 1)$ pixels, since $\sum_{n=1}^{\infty} (\frac{1}{2})^n = 1$. Even though it optimizes a lot on the memory, the pyramid is still very large. If we are blending many images, it would consume a lot of memory. This would intrduce a space-time tradeoff for pre-computing and storing the pyramids.

If the shifts between the images are small, our approach with align them well with sharp outputs. If the shifts are large, the artifacts produced with our method will be similar to the ghosting artifacts in the linear blending. The shifts should be small since the inputs are the reprojected images from the nearby cameras.

5.2 Image Denoising

In the case of NL-means, smilar patches are blended to compute the output patch. The patches may not be well aligned which results in the structures being blurred. In Fig:5.1c, the structures are clearly visible in the difference between the denoised image and the ground truth. We can use our blending approach to blend these patches. We hypothesize that our method would align the input patches and would preserve the structures.

(a) (b) (c)

Figure 5.1: Image denoising : (a) Input image, (b) denoised output, (c) difference between output and the ground truth.

In this case, the size of the patches are very small. The pyramid for individual patches would not require a lot of space. But because of their small size, the number of similar patches corresponding to a specific patch would very large. Computing the pyramids for all these patches during runtime would be very inefficient. In this case, it is not necessary to have many levels of the pyramid. The shifts between the patches can be very small which can be represented using one or two levels. Most of the computation time would involve the construction and the reconstruction of the pyramids. This itself would be very expensive because of the number of patches.

It could be possible to compute the pyramid of the whole image and use patches of the pyramid as an approximation of the true pyramid of the patch. This approximation would be very close if we only use the high frequency levels from the pyramid to compute the phase shift. Since the patches and the shifts between similar patches are small, using only high frequency

bands for the pyramid should be sufficient. The high and low frequency residuals would be averaged like in the original NL-means. By computing the pyramid for the whole image, the local phase of the patches in this pyramid would be used to construct a new pyramid for the output patch after shifting the phases. The new pyramid would be reconstructed to get the output patch. At every patch only one reconstruction of the pyramid would be required to compute the output. This would make the algorithm fast and would make the storage requirements minimal. If we choose o orientations and 2 levels of the pyramid, the space required to store the pyramid would be $M * (2o + 2)$ pixels for an image of size M pixels.

It could be possible that blending the patches after aligning them would align the noise in the patches making the denoising algorithm fail. This should not be the case in general. The noise would appear as noise in the phase signal which would be removed after the median filtering. In addition, since the noise would be random in the patches, it won't be possibe to estimate the shifts between them using local phase. In that case, they would be blended like in the case of linear blending.

— 6 —
Discussion

Phase-based methods can usually handle very small shifts and have been used to blend only two input images. We have developed a phase-based approach which can blend images with larger shifts. It can interpolate between the inputs and can also handle multiple input images. We discuss some details of the approach and possible fulture directions.

The median filter refines the phase difference to preserve the shape of the signal but it also removes very local shift information. If the size of filter is large, very small shifts might not be represented at all. If the size of filter is very small, the peaks in the phase difference are not removed which makes the result bad. We are not sure what size of the filter should give the best output.

In the case of regularization when the threshold is high, there could be some levels where the phase difference was not a good estimate of the shift but it was not regularized. If the threshold is very low, the method ignores information present in the higher freqencies even though they were respresented well. We have observed that keeping the threshold in the range $\pi/6$ to $\pi/4$ produces the best results. It would be important to study other criteria which could capture the quality of the estimate in any level to determine if there is any need for regularization.

We fix the number of orientations of the pyramid to be 5 for all the results. If the number of orientations is very small, the output is less sharp. This is because any band of the pyramid would correspond to a large range of directions. The orientation of the shift would not be represented well. After a certain number, increasing the number of orientations does not visibly

(a) (b)

Figure 6.1: (a) One of the input images, (b) The boat and small waves are well preserved and shifted. The high waves cannot be shifted and is blurred. Image taken from [Meyer et al., 2015].

change the output.

In this method, we do not manipulate the amplitude information at all. In practice, the amplitude has information about structures too. As such, shifting the phases using the same amplitudes often does not produce outputs with similar contrasts. The ringing like artifacts near edges after shifting the phases can be attenuated if the amplitude signal can move to align with the phase. In future, we would like to shift the amplitudes by a rough estimate of the shift so that the image has the same appearance as the input images.

We have seen that the complex steerable pyramids occupy a lot of space which could make it difficult to be used in many applications. Wadhwa et al. [2014] introduced the Riesz pyramid which is less over-complete than the steerable pyramid. At any frequency level, there are two images required to compute the local amplitude, phase and the dominant orientation at each pixel. The local phase of each pixel in each band can be modified to move the pixel in its dominant orientation direction. Since the number of images at any level is only two rather than the number of orientations chosen for the steerable pyramid, the blending can be done using less memory making it easier to be applied for different applications. It would also be faster since the computation and reconstruction of the pyramid is fast.

Since the phase differences have been modified and refined, there is no guarantee that all inputs shifted towards the middle are aligned. The assumption of linearity of phase difference with respect to shifts would not be true in the case of large shifts. In practice, the shifted images are always mis-aligned with very small shifts between them. This introduces some ghosting artifacts or loss of fine structures or specularities in the output. Since we have brought two images with large shifts very close to each other, it could be possible to use other methods like Meyer et al. [2015] on the individual shifted images to get a sharper output.

Bibliography

Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *Computer Vision-ECCV 2004*, pages 25–36. Springer, 2004.

Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65. IEEE, 2005.

Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 425–432. ACM, 2001.

Piotr Didyk, Pitchaya Sitthi-Amorn, William T. Freeman, Frédo Durand, and Wojciech Matusik. Joint view expansion and filtering for automultiscopic 3d displays. *ACM Transactions on Graphics (Proceedings SIGGRAPH Asia 2013, Hong Kong)*, 32(6), 2013.

Martin Eisemann, Bert De Decker, Marcus Magnor, Philippe Bekaert, Edilson De Aguiar, Naveed Ahmed, Christian Theobalt, and Anita Sellent. Floating textures. In *Computer Graphics Forum*, volume 27, pages 409–418. Wiley Online Library, 2008.

David J Fleet and Allan D Jepson. Computation of component image velocity from local phase information. *International journal of computer vision*, 5(1):77–104, 1990.

David J Fleet, Allan D Jepson, and Michael RM Jenkin. Phase-based disparity measurement. *CVGIP: Image understanding*, 53(2):198–210, 1991.

Temujin Gautama and Marc M Van Hulle. A phase-based approach to the estimation of the optical flow field using spatial filtering. *Neural Networks, IEEE Transactions on*, 13(5): 1127–1136, 2002.

Gösta Granlund and Hans Knutsson. Signal processing for computer vision. 1995.

Changil Kim, Henning Zimmer, Yael Pritch, Alexander Sorkine-Hornung, and Markus H Gross. Scene reconstruction from high spatio-angular resolution light fields. *ACM Trans. Graph.*, 32(4):73, 2013.

Simone Meyer, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung. Phase-based frame interpolation for video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1410–1418, 2015.

Bernard Picinbono. On instantaneous amplitude and phase of signals. *Signal Processing, IEEE Transactions on*, 45(3):552–560, 1997.

Terence D Sanger. Stereo disparity computation using gabor filters. *Biological cybernetics*, 59 (6):405–418, 1988.

Eero P Simoncelli and William T Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *icip*, page 3444. IEEE, 1995.

Michael Tao, Jiamin Bai, Pushmeet Kohli, and Sylvain Paris. Simpleflow: A non-iterative, sublinear optical flow algorithm. In *Computer Graphics Forum*, volume 31, pages 345–353. Wiley Online Library, 2012.

Neal Wadhwa, Michael Rubinstein, Frédo Durand, and William T. Freeman. Phase-based video motion processing. *ACM Trans. Graph. (Proceedings SIGGRAPH 2013)*, 32(4), 2013.

Neal Wadhwa, Michael Rubinstein, Frédo Durand, and William T. Freeman. Riesz pyramids for fast phase-based video magnification. 2014.

Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-time video completion. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–120. IEEE, 2004.

Zhoutong Zhang, Yebin Liu, and Qionghai Dai. Light Field from Micro-baseline Image Pair. *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.