Appendix C

Computation of Synthetic Basis Fringes

This section describes two related methods for generating diffraction-specific basis fringes: the iterative constraint method (discussed in the Background section) and a novel simulated annealing approach. These numerical methods are necessitated by the complicated set of spatial and spectral constraints on a given basis fringe. Spatially, the basis fringe must represent a single hogel at a single location in the fringe pattern. It must therefore have a finite width, and must possess homogeneous characteristics across its width (i.e., a virtually constant amplitude). Spectrally, the basis fringe has multiple often conflicting characteristics.

The spectrum of a hogel is divided into sampled increments of width Δ_f. Each basis fringe is responsible for contributing energy in a particular portion of the spectrum and in no others. This corresponds to diffracting light in a finite range of directions. If the spectrum has a bandwidth of BW=0.5 cycles/sample and is divided into N regions of width Δ_f=0.5/N, then each basis fringe i must possess a spectrum that is non-zero in the range $[i\Delta_f,(i+1)\Delta_f]$ for i=[0,N-1]. The shape of the spectrum is a truncated sinc function as required by the sampling theory to recover the continuous spectrum. Other shapes – guassian, rectangular, triangular – were also used to determine their effects on image quality. Each spectrum has a specific width.

An analytic solution to these many constraints is practically impossible. Attempts at deriving closed-form analytical solutions leads to further complexity. More importantly, these analytic approach are mired in interference-based computation. They are prone to the same problems associated with interference-based computation.

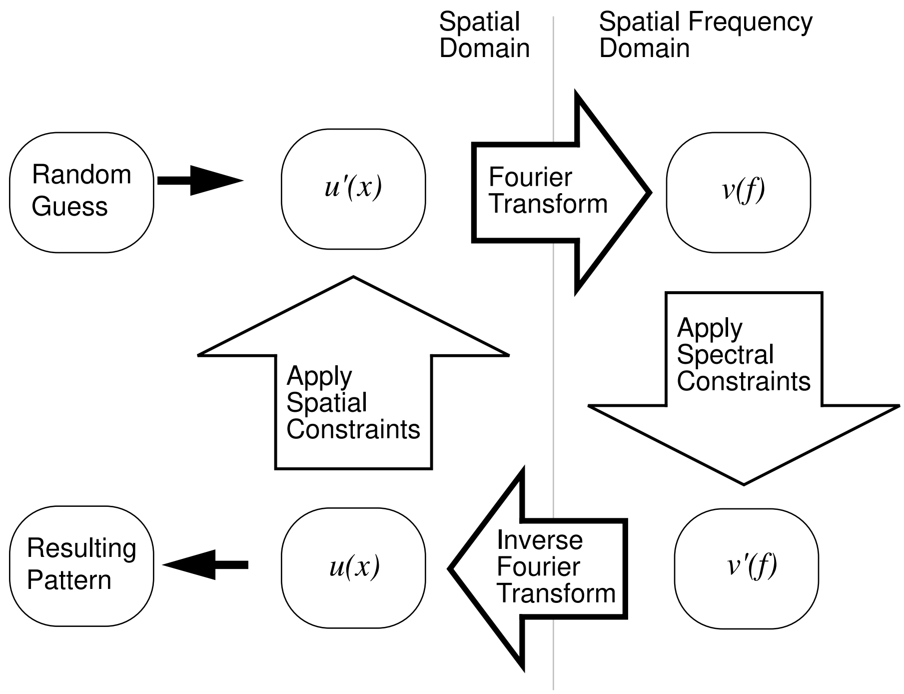
Several numerical methods can be applied to generate basis fringes. These numerical methods produce a fringe that is synthesized from a set of constraints rather than from interference-based analysis. Therefore, these numerically generated fringes are called

synthetic fringes. The numerical methods that worked best for this thesis were the method of iterative constraints, and a novel application of simulated annealing. Both of these algorithms were implemented on the Connection Machine Model 2 with 16K data-parallel processors.

C.1 Method of Iterative Constraints

As discussed in Section 2.7 “Iterative Hologram Computation Methods”on page 34, a typical iterative constraint algorithm involves using the forward and inverse Fourier transforms and alternately applying spatial and spectral constraints. The figure on the next page illustrates this algorithm schematically and in a step by step description.

Algorithm for Method of Iterative Constraints



1. Generate a random fringe pattern of width w.
2. Transform into the spatial frequency domain.
3. Apply the (spectral) constraints on $v(f)$.
4. Inverse transform back to the spatial domain.
5. Apply the (spatial) constraints on $u(x)$.
6. Iterate starting at step 2.

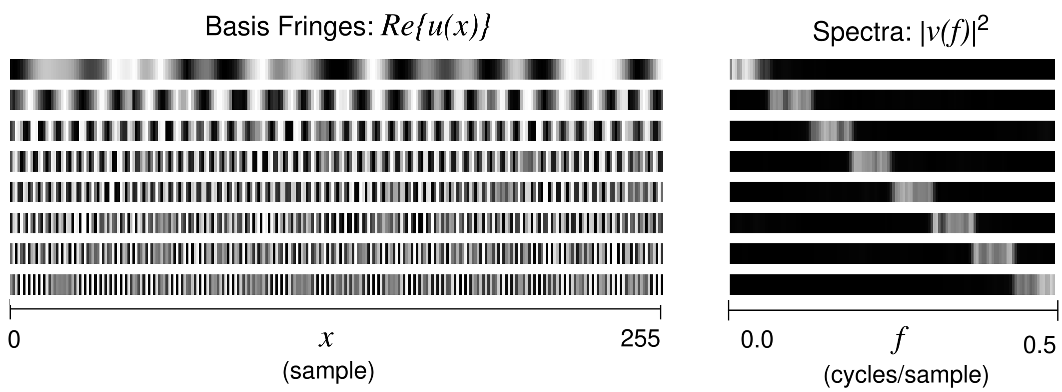
where:

$u(x)$ = computed spatial pattern — $u'(x)$ = modified spatial pattern

$v(f)$ = spectral pattern — $v'(f)$ = modified spectral pattern

As an example of the application of iterative constraints to the synthesis of basis fringes, consider the case of a basis fringe that is to have spectral energy in a rectangular region ranging from 0.0 up to 0.5/8=0.0625 (cycles/sample). The spectral constraint on magnitude is that it be 1.0 within the region [0.0,0.0625] and 0.0 elsewhere. The spatial constraint is that it have a uniform magnitude of 1.0. Phase is unconstrained in both the spatial and spectral domains. For a length of N_h = 256 samples, computation begins with a randomly generated spatial phase and a uniform magnitude of unity. After transforming into the spectral domain, the magnitude is forced to equal the desired spectrum, namely 1.0 in the region [0.0,0.0625] and 0.0 elsewhere. After inverse transforming into the spatial domain, the magnitude is forced to be uniform. This cycle iterates. After 20 iterations, a reasonable solution to the constraints is generally obtained, after which very little changes. This example was run using an iterative constraint algorithm implemented on the CM2. After 100 iterations, the resulting real part of the spatial pattern is shown as the top basis fringe in the following figure:

Eight Basis Fringes



At its right is its spectrum. Included are an additional seven basis fringes, each with a spectrum that is rectangular with width Δ_f=0.5/8=0.0625, each non-zero in the range $[i\Delta_f,(i+1)\Delta_f]$ for i=[0,7], These basis fringes evenly divide the spectrum into eight equal parts. Such a set of basis fringes was used to perform hogel-vector decoding for

the parameters N_h = 256 samples and CR=32. For most cases, the basis fringes used had a gaussian profile, with $1/e^2$ spectral full-width of Δ_f.

The advantage of the iterative constraint algorithm is that it can generate synthetic basis fringes using a wide variety of constraints. It is fast, though speed is not important in precomputation of basis fringes since these basis fringes need only to be computed once and then integrated into diffraction-specific computation algorithms. The more important issue is one of closeness to the desired constraints. Notice that in the example of eight basis fringes shown in the previous figure, the spectra are not perfect rectangles. The ripples and drop-outs in basis fringe spectra lead to speckle-like artifacts in the holographic images. The disadvantage is that this method sometimes stagnates at local minima in the error function[48]. The error function is the difference between the desired spectral characteristics and those in the spectrum of the calculated pattern. One solution was to introduce a small amount of noise into the spatial pattern with each iteration. The amount of noise decreased with each iteration. Still, the synthesized basis fringes were still lacking in spectral quality. The solution was to implement a simulated annealing algorithm.

C.2 Simulated Annealing

Simulated annealing[47] is a numerical optimization algorithm used for a variety of applications. Just as physical annealing (the process of slowly cooling a liquid into a solid) seeks to decrease the global energy state of a system, simulated annealing seeks to minimize the error between certain qualities of a numerical system and their targeted constraints. To further the analogy, simulated annealing sometimes increases error based on a probability function that resembles Boltzmann's probability distribution:

$$Prob\,(E) \sim exp(\frac{-E}{kT}) \qquad \textbf{(C3)}$$

which relates the probability of a system being in a particular energy state E given that its thermal energy is kT. As a physical system cools, it generally moves toward a lower energy state. However, there is always a finite probability that it will move (temporarily) to a higher energy state. In simulated annealing, a random test change is made to the numerical system. Let ΔE be the difference between the system "error" with the test change and the system without it. If the resulting change in error ΔE is negative, then the change is allowed. If ΔE is positive, it may be kept or rejected, depending probabilistically on Equation C3. The system always moves toward a potential decrease in error, but does sometimes move toward a potentially higher error state. This occasional addition of error prevents the system from stagnating at a local minimum (as does the iterative constraints algorithm). The parameter kT must be chosen to allow for occasional increase in error, and must be slowly cooled (reduced) as the system converges to its targeted constraints.

A simulated annealing algorithm was implemented for this thesis on the CM2. The process of randomly altering one sample of a basis function is slow. It is important to begin the annealing with a good guess. Therefore, the basis fringe generated using the method of iterative constraints was used as the initial guess (seed) for the simulated annealing algorithm. The first step in the annealing algorithm is randomly to select a sample of the spatial pattern and to change it, within the spatial constraints. After a test change has been made, the spectral energy is calculated by applying a Fourier transform. The error function after a given iteration was calculated as

$$E = \left\{ \frac{1}{N_h - 1} \cdot \sum_{i=0}^{N_h - 1} \left[|v(f_i)|^2 - |v_c(f_i)|^2 \right] \right\}^{\frac{1}{2}} \qquad \text{(C4)}$$

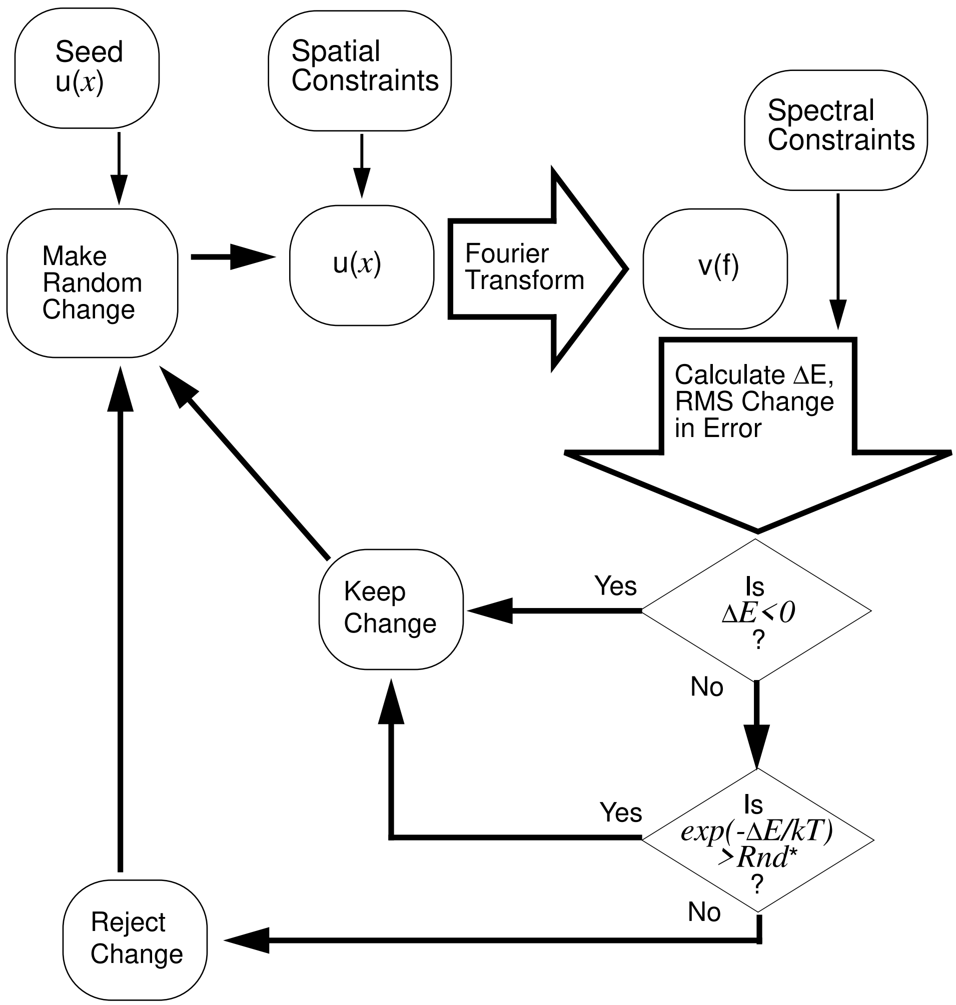
where $v_c(f_i)$ is the targeted spectral magnitude constraint and f_i is the f location of sample i in the basis fringe spectrum. This error function is therefore the root-mean-squared (RMS) energy difference between the targeted spectral constraint and the

spectra of the pattern at each iteration. The simulated annealing algorithm is illustrated on page 166.

With each iteration of the simulated annealing algorithm, the basis fringe sometimes is left unchanged, sometimes moves toward a lower error, and occasionally moves toward a higher error. To illustrate this process, page 167 shows the basis fringe and its spectrum as it progresses generally toward a more satisfactory solution to the constraints.

The combination of the iterative constraint method followed by simulated annealing produced extremely precise basis fringes. The only disadvantage to simulated annealing is that it is slow. However, speed is not important to the precomputation of basis fringes. The advantages of simulated annealing are many. Besides being capable of producing superior basis fringes with tightly constrained spectra, this algorithm has also been applied to the generation of fringes with other constraints. For example, simulated annealing was applied to the synthesis of fringes with highly quantized phase – in some cases binary phase. Results were good, though the cooling schedule must be more carefully engineered in these nonlinear constraint applications. For comparison, in these cases the method of iterative constraints generally stagnated before a satisfactory fringe was generated.

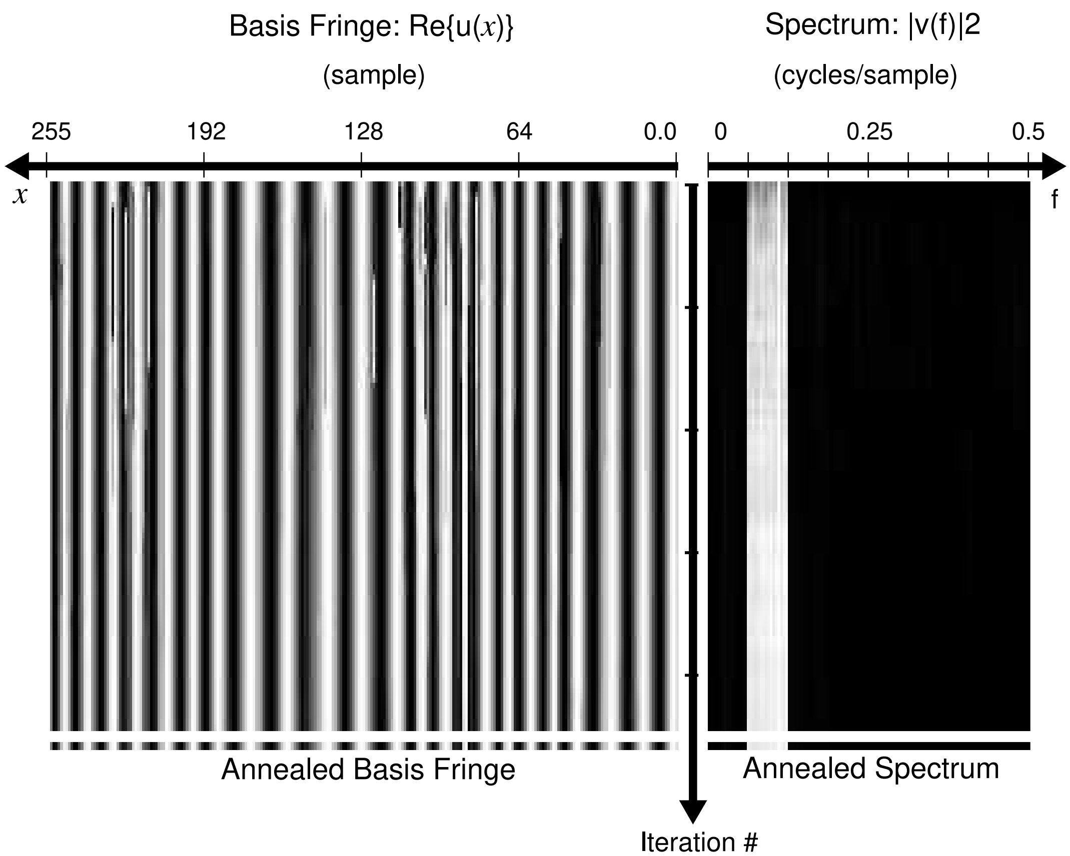
The Simulated Annealing Algorithm



*Rnd = a random number, (0.0,1.0]

This figure shows a schematic of the simulated annealing algorithm as implemented for the synthesis of basis fringes. Essentially, the fringe pattern gradually moves toward its targeted spectral constraints through a probabilistic decision to keep or reject randomly made changes. The annealing precedes until the error function *E* decreased to less than one quantization level per sample.

Simulated Annealing: An Example



This figure shows the process of simulated annealing applied to the synthesis of a basis fringe. At left is the real part of the spatial pattern. At right is the energy of the spectrum. The top line is the initial seed spatial pattern and its spectrum. Each line represents a change that was made and kept. The total number of changes attempted was 10000, and the number that were kept was 440. The bottom line is the annealed basis fringe and its spectrum. This basis fringe was targeted to have a uniform spectrum of 1.0 in the range [0.0625, 0.1250] and no spectral energy outside this range. The initial seed pattern was generated by the method of iterative constraints, which left some deviations from the targeted spectrum (even after 100 iterations). Notice the progression of the spectrum: it began with deviations in the form of dark regions within the region where it was targeted to be uniform. These variations virtually melted away as the annealing progressed. The error function decreased by roughly a factor of 10. Total time for 10000 iterations was two minutes.

Appendix C

Computation of Synthetic Basis Fringes

This section describes two related methods for generating diffraction-specific basis fringes: the iterative constraint method (discussed in the Background section) and a novel simulated annealing approach. These numerical methods are necessitated by the complicated set of spatial and spectral constraints on a given basis fringe. Spatially, the basis fringe must represent a single hogel at a single location in the fringe pattern. It must therefore have a finite width, and must possess homogeneous characteristics across its width (i.e., a virtually constant amplitude). Spectrally, the basis fringe has multiple often conflicting characteristics.

The spectrum of a hogel is divided into sampled increments of width Δ_f. Each basis fringe is responsible for contributing energy in a particular portion of the spectrum and in no others. This corresponds to diffracting light in a finite range of directions. If the spectrum has a bandwidth of BW=0.5 cycles/sample and is divided into N regions of width Δ_f=0.5/N, then each basis fringe i must possess a spectrum that is non-zero in the range $[i\Delta_f,(i+1)\Delta_f]$ for i=[0,N-1]. The shape of the spectrum is a truncated sinc function as required by the sampling theory to recover the continuous spectrum. Other shapes – guassian, rectangular, triangular – were also used to determine their effects on image quality. Each spectrum has a specific width.

An analytic solution to these many constraints is practically impossible. Attempts at deriving closed-form analytical solutions leads to further complexity. More importantly, these analytic approach are mired in interference-based computation. They are prone to the same problems associated with interference-based computation.

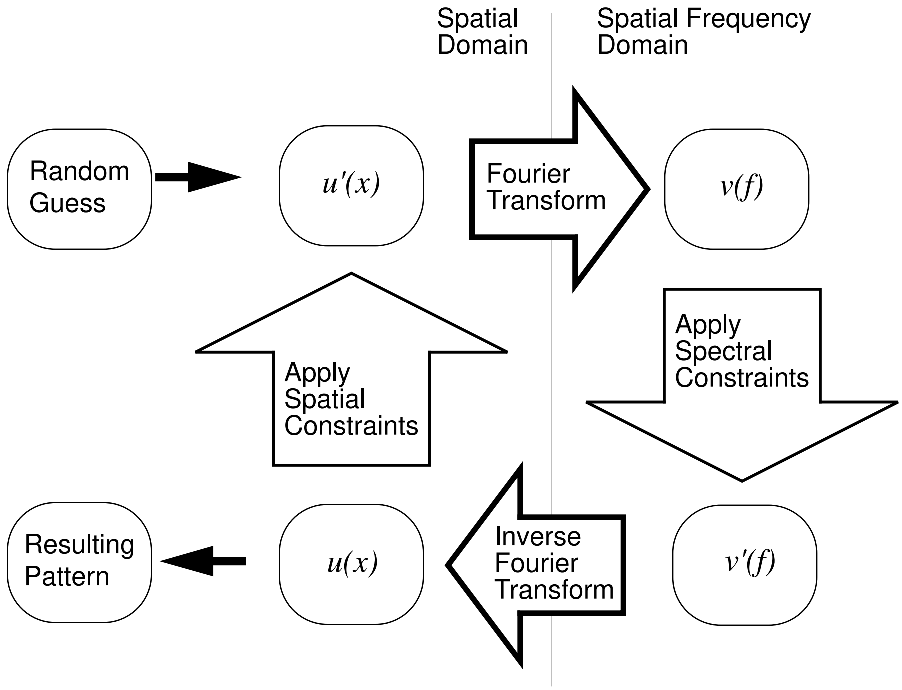
Several numerical methods can be applied to generate basis fringes. These numerical methods produce a fringe that is synthesized from a set of constraints rather than from interference-based analysis. Therefore, these numerically generated fringes are called

synthetic fringes. The numerical methods that worked best for this thesis were the method of iterative constraints, and a novel application of simulated annealing. Both of these algorithms were implemented on the Connection Machine Model 2 with 16K data-parallel processors.

C.1 Method of Iterative Constraints

As discussed in Section 2.7 “Iterative Hologram Computation Methods”on page 34, a typical iterative constraint algorithm involves using the forward and inverse Fourier transforms and alternately applying spatial and spectral constraints. The figure on the next page illustrates this algorithm schematically and in a step by step description.

Algorithm for Method of Iterative Constraints



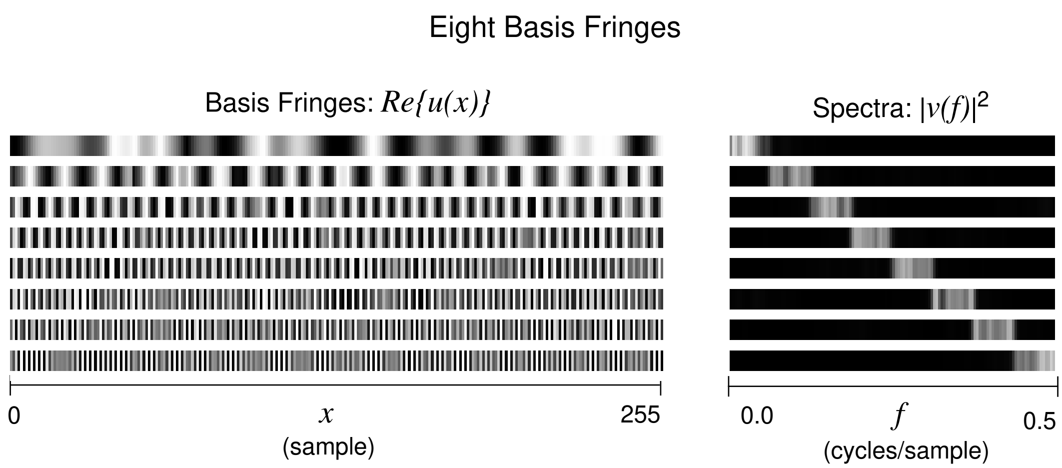
1. Generate a random fringe pattern of width w.
2. Transform into the spatial frequency domain.
3. Apply the (spectral) constraints on $v(f)$.
4. Inverse transform back to the spatial domain.
5. Apply the (spatial) constraints on $u(x)$.
6. Iterate starting at step 2.

where:

$u(x)$ = computed spatial pattern $u'(x)$ = modified spatial pattern

$v(f)$ = spectral pattern $v'(f)$ = modified spectral pattern

As an example of the application of iterative constraints to the synthesis of basis fringes, consider the case of a basis fringe that is to have spectral energy in a rectangular region ranging from 0.0 up to 0.5/8=0.0625 (cycles/sample). The spectral constraint on magnitude is that it be 1.0 within the region [0.0,0.0625] and 0.0 elsewhere. The spatial constraint is that it have a uniform magnitude of 1.0. Phase is unconstrained in both the spatial and spectral domains. For a length of N_h = 256 samples, computation begins with a randomly generated spatial phase and a uniform magnitude of unity. After transforming into the spectral domain, the magnitude is forced to equal the desired spectrum, namely 1.0 in the region [0.0,0.0625] and 0.0 elsewhere. After inverse transforming into the spatial domain, the magnitude is forced to be uniform. This cycle iterates. After 20 iterations, a reasonable solution to the constraints is generally obtained, after which very little changes. This example was run using an iterative constraint algorithm implemented on the CM2. After 100 iterations, the resulting real part of the spatial pattern is shown as the top basis fringe in the following figure:



At its right is its spectrum. Included are an additional seven basis fringes, each with a spectrum that is rectangular with width Δ_f=0.5/8=0.0625, each non-zero in the range [$i\Delta_f$,$(i+1)\Delta_f$] for i=[0,7], These basis fringes evenly divide the spectrum into eight equal parts. Such a set of basis fringes was used to perform hogel-vector decoding for

the parameters N_h = 256 samples and CR=32. For most cases, the basis fringes used had a gaussian profile, with $1/e^2$ spectral full-width of Δ_f.

The advantage of the iterative constraint algorithm is that it can generate synthetic basis fringes using a wide variety of constraints. It is fast, though speed is not important in precomputation of basis fringes since these basis fringes need only to be computed once and then integrated into diffraction-specific computation algorithms. The more important issue is one of closeness to the desired constraints. Notice that in the example of eight basis fringes shown in the previous figure, the spectra are not perfect rectangles. The ripples and drop-outs in basis fringe spectra lead to speckle-like artifacts in the holographic images. The disadvantage is that this method sometimes stagnates at local minima in the error function[48]. The error function is the difference between the desired spectral characteristics and those in the spectrum of the calculated pattern. One solution was to introduce a small amount of noise into the spatial pattern with each iteration. The amount of noise decreased with each iteration. Still, the synthesized basis fringes were still lacking in spectral quality. The solution was to implement a simulated annealing algorithm.

C.2 Simulated Annealing

Simulated annealing[47] is a numerical optimization algorithm used for a variety of applications. Just as physical annealing (the process of slowly cooling a liquid into a solid) seeks to decrease the global energy state of a system, simulated annealing seeks to minimize the error between certain qualities of a numerical system and their targeted constraints. To further the analogy, simulated annealing sometimes increases error based on a probability function that resembles Boltzmann's probability distribution:

$$Prob(E) \sim exp\left(\frac{-E}{kT}\right) \quad \text{(C3)}$$

which relates the probability of a system being in a particular energy state E given that its thermal energy is kT. As a physical system cools, it generally moves toward a lower energy state. However, there is always a finite probability that it will move (temporarily) to a higher energy state. In simulated annealing, a random test change is made to the numerical system. Let ΔE be the difference between the system “error” with the test change and the system without it. If the resulting change in error ΔE is negative, then the change is allowed. If ΔE is positive, it may be kept or rejected, depending probabilistically on Equation C3. The system always moves toward a potential decrease in error, but does sometimes move toward a potentially higher error state. This occasional addition of error prevents the system from stagnating at a local minimum (as does the iterative constraints algorithm). The parameter kT must be chosen to allow for occasional increase in error, and must be slowly cooled (reduced) as the system converges to its targeted constraints.

A simulated annealing algorithm was implemented for this thesis on the CM2. The process of randomly altering one sample of a basis function is slow. It is important to begin the annealing with a good guess. Therefore, the basis fringe generated using the method of iterative constraints was used as the initial guess (seed) for the simulated annealing algorithm. The first step in the annealing algorithm is randomly to select a sample of the spatial pattern and to change it, within the spatial constraints. After a test change has been made, the spectral energy is calculated by applying a Fourier transform. The error function after a given iteration was calculated as

$$E = \left\{ \frac{1}{N_h - 1} \cdot \sum_{i=0}^{N_h - 1} \left[|v(f_i)|^2 - |v_c(f_i)|^2 \right] \right\}^{\frac{1}{2}} \qquad \text{(C4)}$$

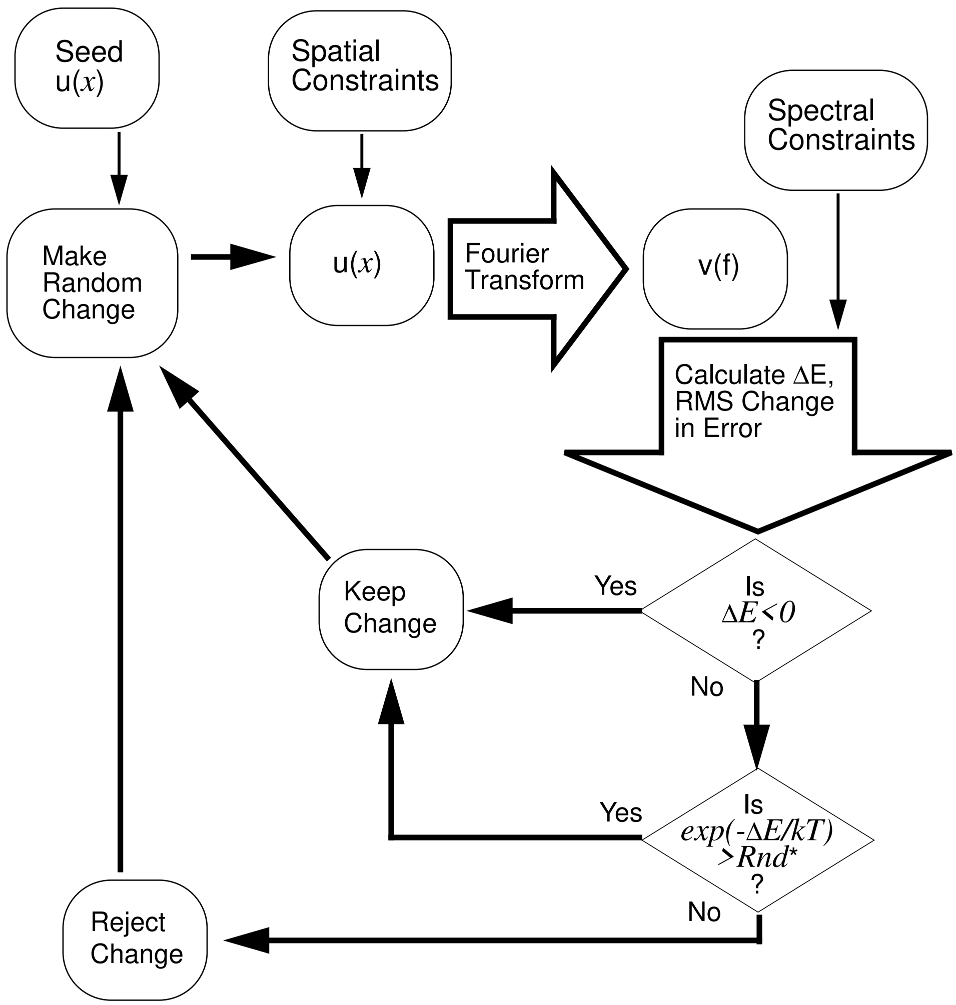
where $v_c(f_i)$ is the targeted spectral magnitude constraint and f_i is the f location of sample i in the basis fringe spectrum. This error function is therefore the root-mean-squared (RMS) energy difference between the targeted spectral constraint and the

spectra of the pattern at each iteration. The simulated annealing algorithm is illustrated on page 166.

With each iteration of the simulated annealing algorithm, the basis fringe sometimes is left unchanged, sometimes moves toward a lower error, and occasionally moves toward a higher error. To illustrate this process, page 167 shows the basis fringe and its spectrum as it progresses generally toward a more satisfactory solution to the constraints.

The combination of the iterative constraint method followed by simulated annealing produced extremely precise basis fringes. The only disadvantage to simulated annealing is that it is slow. However, speed is not important to the precomputation of basis fringes. The advantages of simulated annealing are many. Besides being capable of producing superior basis fringes with tightly constrained spectra, this algorithm has also been applied to the generation of fringes with other constraints. For example, simulated annealing was applied to the synthesis of fringes with highly quantized phase – in some cases binary phase. Results were good, though the cooling schedule must be more carefully engineered in these nonlinear constraint applications. For comparison, in these cases the method of iterative constraints generally stagnated before a satisfactory fringe was generated.

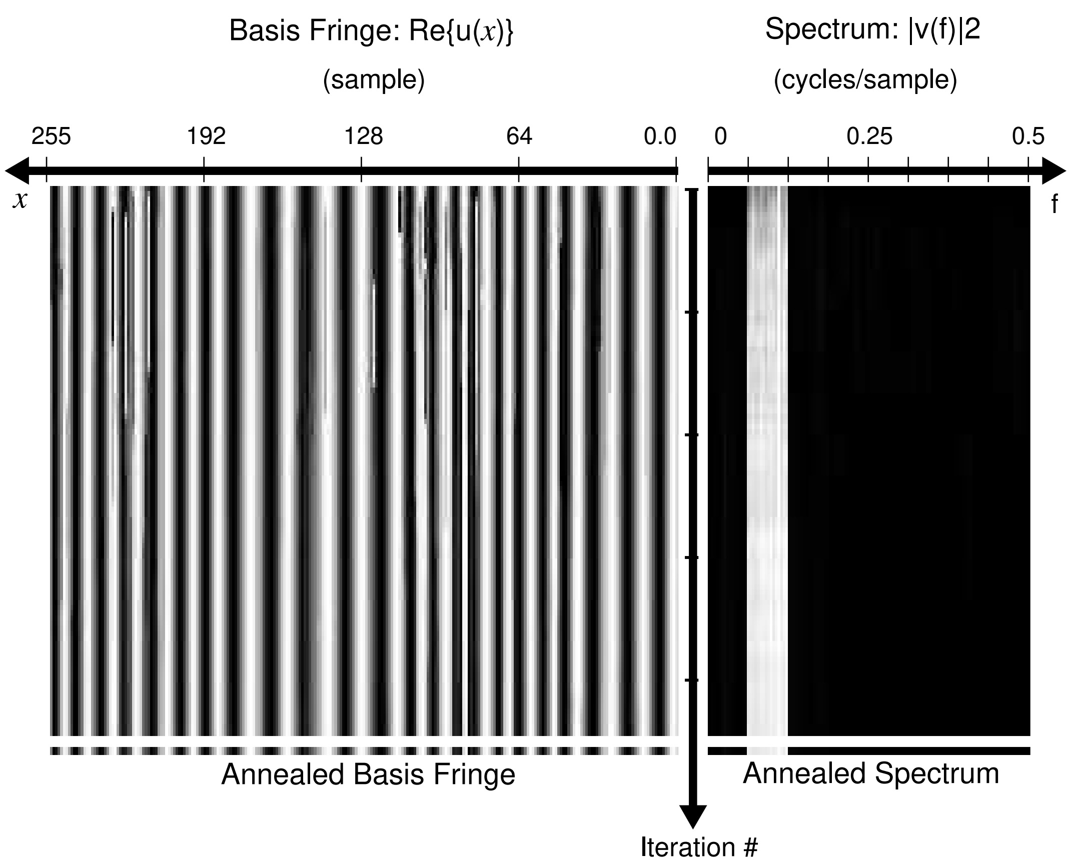
The Simulated Annealing Algorithm



*Rnd = a random number, (0.0,1.0]

This figure shows a schematic of the simulated annealing algorithm as implemented for the synthesis of basis fringes. Essentially, the fringe pattern gradually moves toward its targeted spectral constraints through a probabilistic decision to keep or reject randomly made changes. The annealing precedes until the error function *E* decreased to less than one quantization level per sample.

Simulated Annealing: An Example



This figure shows the process of simulated annealing applied to the synthesis of a basis fringe. At left is the real part of the spatial pattern. At right is the energy of the spectrum. The top line is the initial seed spatial pattern and its spectrum. Each line represents a change that was made and kept. The total number of changes attempted was 10000, and the number that were kept was 440. The bottom line is the annealed basis fringe and its spectrum. This basis fringe was targeted to have a uniform spectrum of 1.0 in the range [0.0625, 0.1250] and no spectral energy outside this range. The initial seed pattern was generated by the method of iterative constraints, which left some deviations from the targeted spectrum (even after 100 iterations). Notice the progression of the spectrum: it began with deviations in the form of dark regions within the region where it was targeted to be uniform. These variations virtually melted away as the annealing progressed. The error function decreased by roughly a factor of 10. Total time for 10000 iterations was two minutes.

Appendix C

Computation of Synthetic Basis Fringes

This section describes two related methods for generating diffraction-specific basis fringes: the iterative constraint method (discussed in the Background section) and a novel simulated annealing approach. These numerical methods are necessitated by the complicated set of spatial and spectral constraints on a given basis fringe. Spatially, the basis fringe must represent a single hogel at a single location in the fringe pattern. It must therefore have a finite width, and must possess homogeneous characteristics across its width (i.e., a virtually constant amplitude). Spectrally, the basis fringe has multiple often conflicting characteristics.

The spectrum of a hogel is divided into sampled increments of width Δ_f. Each basis fringe is responsible for contributing energy in a particular portion of the spectrum and in no others. This corresponds to diffracting light in a finite range of directions. If the spectrum has a bandwidth of BW=0.5 cycles/sample and is divided into N regions of width Δ_f=0.5/N, then each basis fringe i must possess a spectrum that is non-zero in the range $[i\Delta_f,(i+1)\Delta_f]$ for i=[0,N-1]. The shape of the spectrum is a truncated sinc function as required by the sampling theory to recover the continuous spectrum. Other shapes – guassian, rectangular, triangular – were also used to determine their effects on image quality. Each spectrum has a specific width.

An analytic solution to these many constraints is practically impossible. Attempts at deriving closed-form analytical solutions leads to further complexity. More importantly, these analytic approach are mired in interference-based computation. They are prone to the same problems associated with interference-based computation.

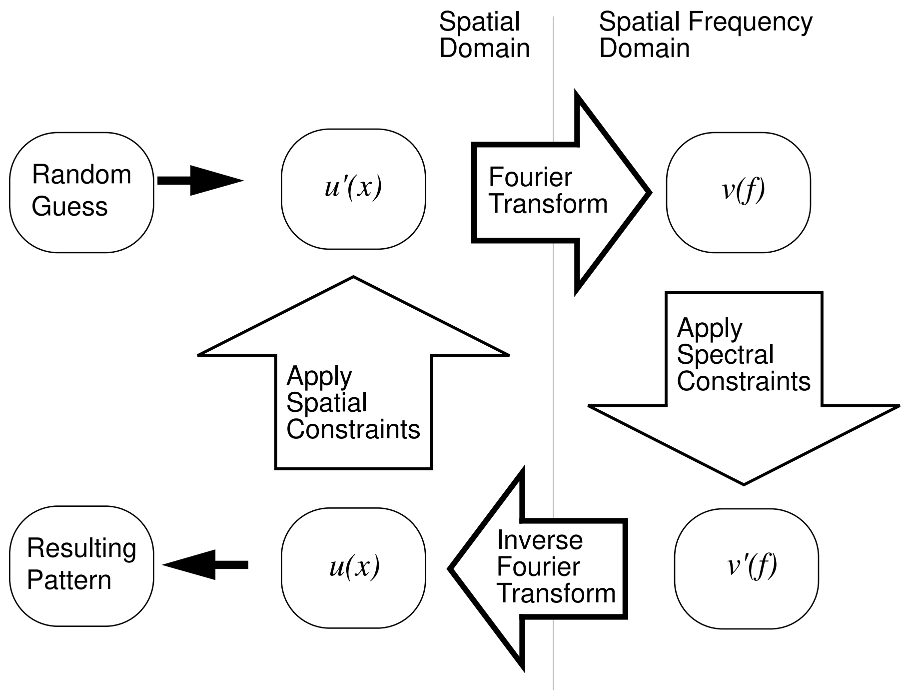
Several numerical methods can be applied to generate basis fringes. These numerical methods produce a fringe that is synthesized from a set of constraints rather than from interference-based analysis. Therefore, these numerically generated fringes are called

synthetic fringes. The numerical methods that worked best for this thesis were the method of iterative constraints, and a novel application of simulated annealing. Both of these algorithms were implemented on the Connection Machine Model 2 with 16K data-parallel processors.

C.1 Method of Iterative Constraints

As discussed in Section 2.7 “Iterative Hologram Computation Methods”on page 34, a typical iterative constraint algorithm involves using the forward and inverse Fourier transforms and alternately applying spatial and spectral constraints. The figure on the next page illustrates this algorithm schematically and in a step by step description.

Algorithm for Method of Iterative Constraints



1. Generate a random fringe pattern of width w.
2. Transform into the spatial frequency domain.
3. Apply the (spectral) constraints on $v(f)$.
4. Inverse transform back to the spatial domain.
5. Apply the (spatial) constraints on $u(x)$.
6. Iterate starting at step 2.

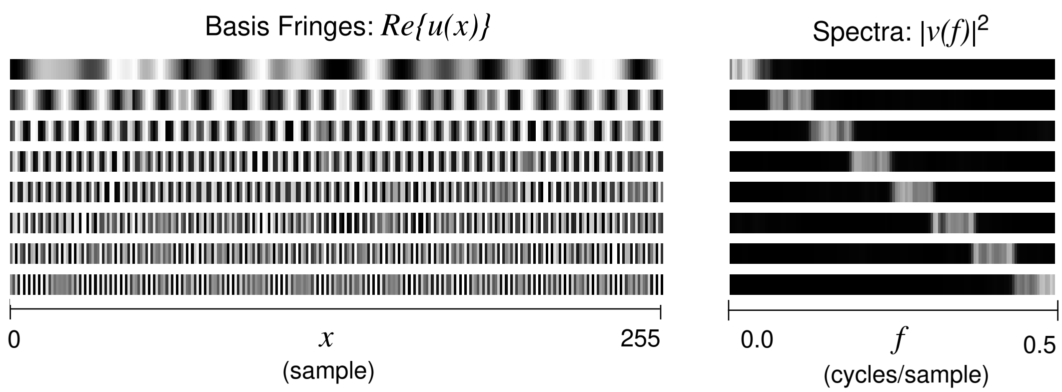
where:

$u(x)$ = computed spatial pattern — $u'(x)$ = modified spatial pattern

$v(f)$ = spectral pattern — $v'(f)$ = modified spectral pattern

As an example of the application of iterative constraints to the synthesis of basis fringes, consider the case of a basis fringe that is to have spectral energy in a rectangular region ranging from 0.0 up to 0.5/8=0.0625 (cycles/sample). The spectral constraint on magnitude is that it be 1.0 within the region [0.0,0.0625] and 0.0 elsewhere. The spatial constraint is that it have a uniform magnitude of 1.0. Phase is unconstrained in both the spatial and spectral domains. For a length of N_h = 256 samples, computation begins with a randomly generated spatial phase and a uniform magnitude of unity. After transforming into the spectral domain, the magnitude is forced to equal the desired spectrum, namely 1.0 in the region [0.0,0.0625] and 0.0 elsewhere. After inverse transforming into the spatial domain, the magnitude is forced to be uniform. This cycle iterates. After 20 iterations, a reasonable solution to the constraints is generally obtained, after which very little changes. This example was run using an iterative constraint algorithm implemented on the CM2. After 100 iterations, the resulting real part of the spatial pattern is shown as the top basis fringe in the following figure:

Eight Basis Fringes



At its right is its spectrum. Included are an additional seven basis fringes, each with a spectrum that is rectangular with width Δ_f=0.5/8=0.0625, each non-zero in the range [$i\Delta_f$,$(i+1)\Delta_f$] for i=[0,7], These basis fringes evenly divide the spectrum into eight equal parts. Such a set of basis fringes was used to perform hogel-vector decoding for

the parameters N_h = 256 samples and CR=32. For most cases, the basis fringes used had a gaussian profile, with $1/e^2$ spectral full-width of Δ_f.

The advantage of the iterative constraint algorithm is that it can generate synthetic basis fringes using a wide variety of constraints. It is fast, though speed is not important in precomputation of basis fringes since these basis fringes need only to be computed once and then integrated into diffraction-specific computation algorithms. The more important issue is one of closeness to the desired constraints. Notice that in the example of eight basis fringes shown in the previous figure, the spectra are not perfect rectangles. The ripples and drop-outs in basis fringe spectra lead to speckle-like artifacts in the holographic images. The disadvantage is that this method sometimes stagnates at local minima in the error function[48]. The error function is the difference between the desired spectral characteristics and those in the spectrum of the calculated pattern. One solution was to introduce a small amount of noise into the spatial pattern with each iteration. The amount of noise decreased with each iteration. Still, the synthesized basis fringes were still lacking in spectral quality. The solution was to implement a simulated annealing algorithm.

C.2 Simulated Annealing

Simulated annealing[47] is a numerical optimization algorithm used for a variety of applications. Just as physical annealing (the process of slowly cooling a liquid into a solid) seeks to decrease the global energy state of a system, simulated annealing seeks to minimize the error between certain qualities of a numerical system and their targeted constraints. To further the analogy, simulated annealing sometimes increases error based on a probability function that resembles Boltzmann's probability distribution:

$$Prob\,(E) \sim exp\left(\frac{-E}{kT}\right) \quad \text{(C3)}$$

which relates the probability of a system being in a particular energy state E given that its thermal energy is kT. As a physical system cools, it generally moves toward a lower energy state. However, there is always a finite probability that it will move (temporarily) to a higher energy state. In simulated annealing, a random test change is made to the numerical system. Let ΔE be the difference between the system "error" with the test change and the system without it. If the resulting change in error ΔE is negative, then the change is allowed. If ΔE is positive, it may be kept or rejected, depending probabilistically on Equation C3. The system always moves toward a potential decrease in error, but does sometimes move toward a potentially higher error state. This occasional addition of error prevents the system from stagnating at a local minimum (as does the iterative constraints algorithm). The parameter kT must be chosen to allow for occasional increase in error, and must be slowly cooled (reduced) as the system converges to its targeted constraints.

A simulated annealing algorithm was implemented for this thesis on the CM2. The process of randomly altering one sample of a basis function is slow. It is important to begin the annealing with a good guess. Therefore, the basis fringe generated using the method of iterative constraints was used as the initial guess (seed) for the simulated annealing algorithm. The first step in the annealing algorithm is randomly to select a sample of the spatial pattern and to change it, within the spatial constraints. After a test change has been made, the spectral energy is calculated by applying a Fourier transform. The error function after a given iteration was calculated as

$$E = \left\{ \frac{1}{N_h - 1} \cdot \sum_{i=0}^{N_h - 1} \left[|v(f_i)|^2 - |v_c(f_i)|^2 \right] \right\}^{\frac{1}{2}} \tag{C4}$$

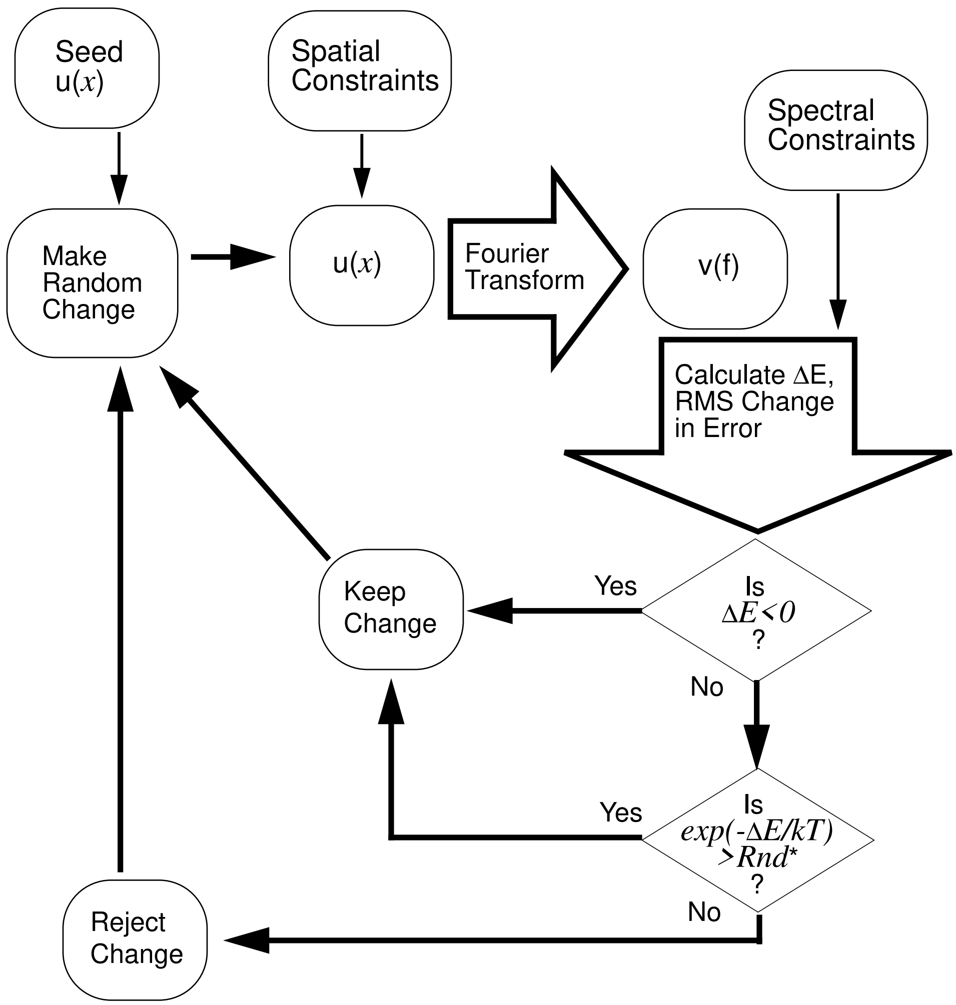
where $v_c(f_i)$ is the targeted spectral magnitude constraint and f_i is the f location of sample i in the basis fringe spectrum. This error function is therefore the root-mean-squared (RMS) energy difference between the targeted spectral constraint and the

spectra of the pattern at each iteration. The simulated annealing algorithm is illustrated on page 166.

With each iteration of the simulated annealing algorithm, the basis fringe sometimes is left unchanged, sometimes moves toward a lower error, and occasionally moves toward a higher error. To illustrate this process, page 167 shows the basis fringe and its spectrum as it progresses generally toward a more satisfactory solution to the constraints.

The combination of the iterative constraint method followed by simulated annealing produced extremely precise basis fringes. The only disadvantage to simulated annealing is that it is slow. However, speed is not important to the precomputation of basis fringes. The advantages of simulated annealing are many. Besides being capable of producing superior basis fringes with tightly constrained spectra, this algorithm has also been applied to the generation of fringes with other constraints. For example, simulated annealing was applied to the synthesis of fringes with highly quantized phase – in some cases binary phase. Results were good, though the cooling schedule must be more carefully engineered in these nonlinear constraint applications. For comparison, in these cases the method of iterative constraints generally stagnated before a satisfactory fringe was generated.

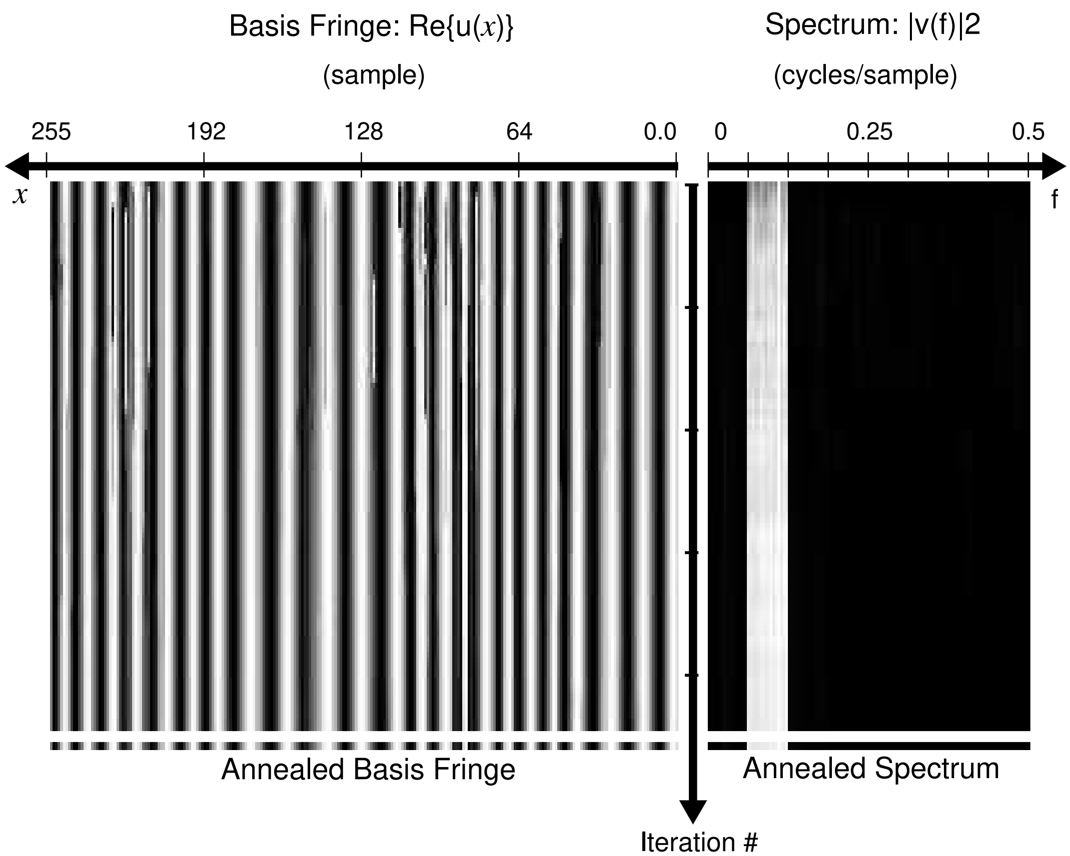
The Simulated Annealing Algorithm



*Rnd = a random number, (0.0,1.0]

This figure shows a schematic of the simulated annealing algorithm as implemented for the synthesis of basis fringes. Essentially, the fringe pattern gradually moves toward its targeted spectral constraints through a probabilistic decision to keep or reject randomly made changes. The annealing precedes until the error function *E* decreased to less than one quantization level per sample.

Simulated Annealing: An Example



This figure shows the process of simulated annealing applied to the synthesis of a basis fringe. At left is the real part of the spatial pattern. At right is the energy of the spectrum. The top line is the initial seed spatial pattern and its spectrum. Each line represents a change that was made and kept. The total number of changes attempted was 10000, and the number that were kept was 440. The bottom line is the annealed basis fringe and its spectrum. This basis fringe was targeted to have a uniform spectrum of 1.0 in the range [0.0625, 0.1250] and no spectral energy outside this range. The initial seed pattern was generated by the method of iterative constraints, which left some deviations from the targeted spectrum (even after 100 iterations). Notice the progression of the spectrum: it began with deviations in the form of dark regions within the region where it was targeted to be uniform. These variations virtually melted away as the annealing progressed. The error function decreased by roughly a factor of 10. Total time for 10000 iterations was two minutes.

Appendix C

Computation of Synthetic Basis Fringes

This section describes two related methods for generating diffraction-specific basis fringes: the iterative constraint method (discussed in the Background section) and a novel simulated annealing approach. These numerical methods are necessitated by the complicated set of spatial and spectral constraints on a given basis fringe. Spatially, the basis fringe must represent a single hogel at a single location in the fringe pattern. It must therefore have a finite width, and must possess homogeneous characteristics across its width (i.e., a virtually constant amplitude). Spectrally, the basis fringe has multiple often conflicting characteristics.

The spectrum of a hogel is divided into sampled increments of width Δ_f. Each basis fringe is responsible for contributing energy in a particular portion of the spectrum and in no others. This corresponds to diffracting light in a finite range of directions. If the spectrum has a bandwidth of BW=0.5 cycles/sample and is divided into N regions of width Δ_f=0.5/N, then each basis fringe i must possess a spectrum that is non-zero in the range [$i\Delta_f$,$(i+1)\Delta_f$] for i=[0,N-1]. The shape of the spectrum is a truncated sinc function as required by the sampling theory to recover the continuous spectrum. Other shapes – guassian, rectangular, triangular – were also used to determine their effects on image quality. Each spectrum has a specific width.

An analytic solution to these many constraints is practically impossible. Attempts at deriving closed-form analytical solutions leads to further complexity. More importantly, these analytic approach are mired in interference-based computation. They are prone to the same problems associated with interference-based computation.

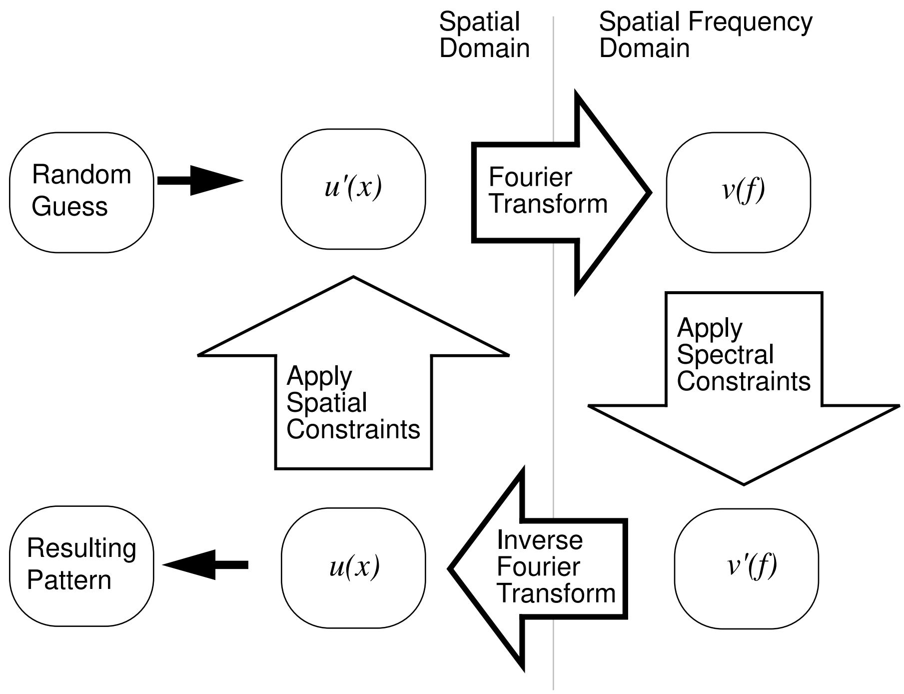
Several numerical methods can be applied to generate basis fringes. These numerical methods produce a fringe that is synthesized from a set of constraints rather than from interference-based analysis. Therefore, these numerically generated fringes are called

synthetic fringes. The numerical methods that worked best for this thesis were the method of iterative constraints, and a novel application of simulated annealing. Both of these algorithms were implemented on the Connection Machine Model 2 with 16K data-parallel processors.

C.1 Method of Iterative Constraints

As discussed in Section 2.7 "Iterative Hologram Computation Methods"on page 34, a typical iterative constraint algorithm involves using the forward and inverse Fourier transforms and alternately applying spatial and spectral constraints. The figure on the next page illustrates this algorithm schematically and in a step by step description.

Algorithm for Method of Iterative Constraints



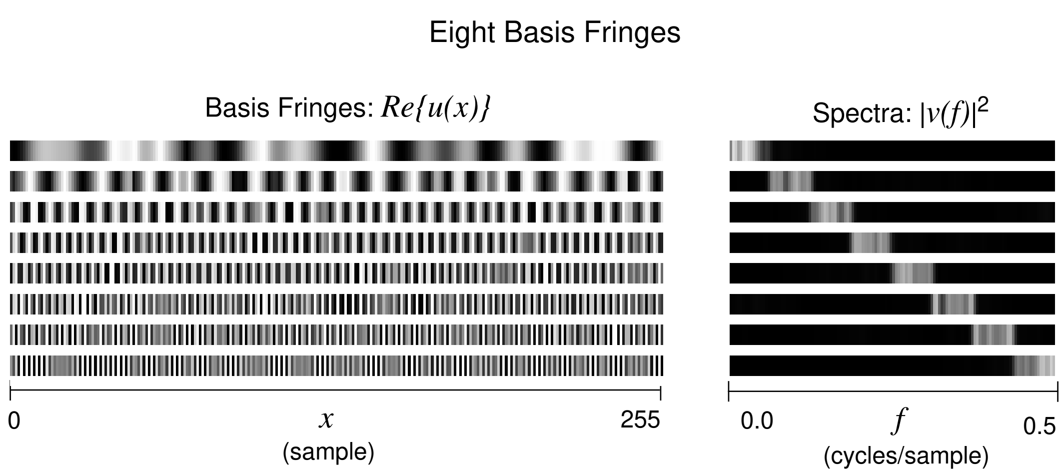
1. Generate a random fringe pattern of width w.
2. Transform into the spatial frequency domain.
3. Apply the (spectral) constraints on $v(f)$.
4. Inverse transform back to the spatial domain.
5. Apply the (spatial) constraints on $u(x)$.
6. Iterate starting at step 2.

where:

$u(x)$ = computed spatial pattern $u'(x)$ = modified spatial pattern

$v(f)$ = spectral pattern $v'(f)$ = modified spectral pattern

As an example of the application of iterative constraints to the synthesis of basis fringes, consider the case of a basis fringe that is to have spectral energy in a rectangular region ranging from 0.0 up to 0.5/8=0.0625 (cycles/sample). The spectral constraint on magnitude is that it be 1.0 within the region [0.0,0.0625] and 0.0 elsewhere. The spatial constraint is that it have a uniform magnitude of 1.0. Phase is unconstrained in both the spatial and spectral domains. For a length of N_h = 256 samples, computation begins with a randomly generated spatial phase and a uniform magnitude of unity. After transforming into the spectral domain, the magnitude is forced to equal the desired spectrum, namely 1.0 in the region [0.0,0.0625] and 0.0 elsewhere. After inverse transforming into the spatial domain, the magnitude is forced to be uniform. This cycle iterates. After 20 iterations, a reasonable solution to the constraints is generally obtained, after which very little changes. This example was run using an iterative constraint algorithm implemented on the CM2. After 100 iterations, the resulting real part of the spatial pattern is shown as the top basis fringe in the following figure:



At its right is its spectrum. Included are an additional seven basis fringes, each with a spectrum that is rectangular with width Δ_f=0.5/8=0.0625, each non-zero in the range $[i\Delta_f,(i+1)\Delta_f]$ for i=[0,7], These basis fringes evenly divide the spectrum into eight equal parts. Such a set of basis fringes was used to perform hogel-vector decoding for

the parameters N_h = 256 samples and CR=32. For most cases, the basis fringes used had a gaussian profile, with $1/e^2$ spectral full-width of Δ_f.

The advantage of the iterative constraint algorithm is that it can generate synthetic basis fringes using a wide variety of constraints. It is fast, though speed is not important in precomputation of basis fringes since these basis fringes need only to be computed once and then integrated into diffraction-specific computation algorithms. The more important issue is one of closeness to the desired constraints. Notice that in the example of eight basis fringes shown in the previous figure, the spectra are not perfect rectangles. The ripples and drop-outs in basis fringe spectra lead to speckle-like artifacts in the holographic images. The disadvantage is that this method sometimes stagnates at local minima in the error function[48]. The error function is the difference between the desired spectral characteristics and those in the spectrum of the calculated pattern. One solution was to introduce a small amount of noise into the spatial pattern with each iteration. The amount of noise decreased with each iteration. Still, the synthesized basis fringes were still lacking in spectral quality. The solution was to implement a simulated annealing algorithm.

C.2 Simulated Annealing

Simulated annealing[47] is a numerical optimization algorithm used for a variety of applications. Just as physical annealing (the process of slowly cooling a liquid into a solid) seeks to decrease the global energy state of a system, simulated annealing seeks to minimize the error between certain qualities of a numerical system and their targeted constraints. To further the analogy, simulated annealing sometimes increases error based on a probability function that resembles Boltzmann's probability distribution:

$$Prob(E) \sim exp\left(\frac{-E}{kT}\right) \tag{C3}$$

which relates the probability of a system being in a particular energy state E given that its thermal energy is kT. As a physical system cools, it generally moves toward a lower energy state. However, there is always a finite probability that it will move (temporarily) to a higher energy state. In simulated annealing, a random test change is made to the numerical system. Let ΔE be the difference between the system "error" with the test change and the system without it. If the resulting change in error ΔE is negative, then the change is allowed. If ΔE is positive, it may be kept or rejected, depending probabilistically on Equation C3. The system always moves toward a potential decrease in error, but does sometimes move toward a potentially higher error state. This occasional addition of error prevents the system from stagnating at a local minimum (as does the iterative constraints algorithm). The parameter kT must be chosen to allow for occasional increase in error, and must be slowly cooled (reduced) as the system converges to its targeted constraints.

A simulated annealing algorithm was implemented for this thesis on the CM2. The process of randomly altering one sample of a basis function is slow. It is important to begin the annealing with a good guess. Therefore, the basis fringe generated using the method of iterative constraints was used as the initial guess (seed) for the simulated annealing algorithm. The first step in the annealing algorithm is randomly to select a sample of the spatial pattern and to change it, within the spatial constraints. After a test change has been made, the spectral energy is calculated by applying a Fourier transform. The error function after a given iteration was calculated as

$$E = \left\{ \frac{1}{N_h - 1} \cdot \sum_{i=0}^{N_h - 1} \left[|v(f_i)|^2 - |v_c(f_i)|^2 \right] \right\}^{\frac{1}{2}} \tag{C4}$$

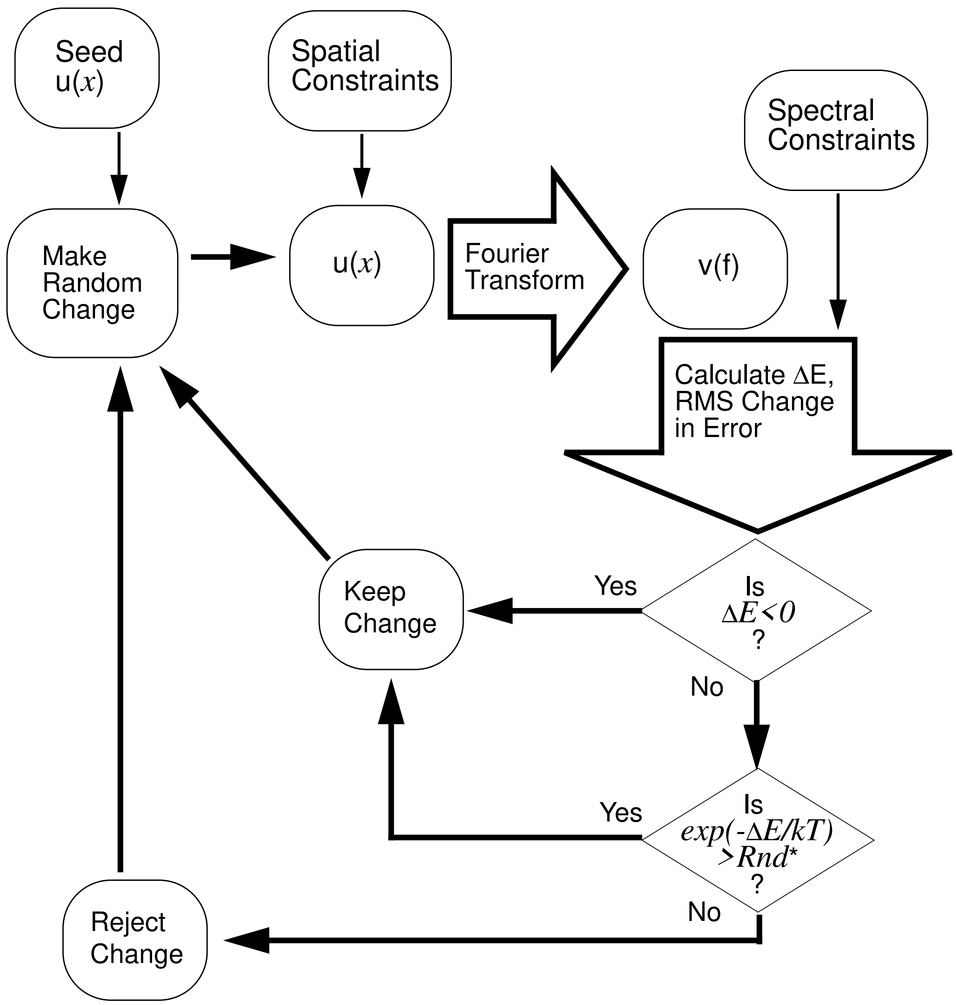
where $v_c(f_i)$ is the targeted spectral magnitude constraint and f_i is the f location of sample i in the basis fringe spectrum. This error function is therefore the root-mean-squared (RMS) energy difference between the targeted spectral constraint and the

spectra of the pattern at each iteration. The simulated annealing algorithm is illustrated on page 166.

With each iteration of the simulated annealing algorithm, the basis fringe sometimes is left unchanged, sometimes moves toward a lower error, and occasionally moves toward a higher error. To illustrate this process, page 167 shows the basis fringe and its spectrum as it progresses generally toward a more satisfactory solution to the constraints.

The combination of the iterative constraint method followed by simulated annealing produced extremely precise basis fringes. The only disadvantage to simulated annealing is that it is slow. However, speed is not important to the precomputation of basis fringes. The advantages of simulated annealing are many. Besides being capable of producing superior basis fringes with tightly constrained spectra, this algorithm has also been applied to the generation of fringes with other constraints. For example, simulated annealing was applied to the synthesis of fringes with highly quantized phase – in some cases binary phase. Results were good, though the cooling schedule must be more carefully engineered in these nonlinear constraint applications. For comparison, in these cases the method of iterative constraints generally stagnated before a satisfactory fringe was generated.

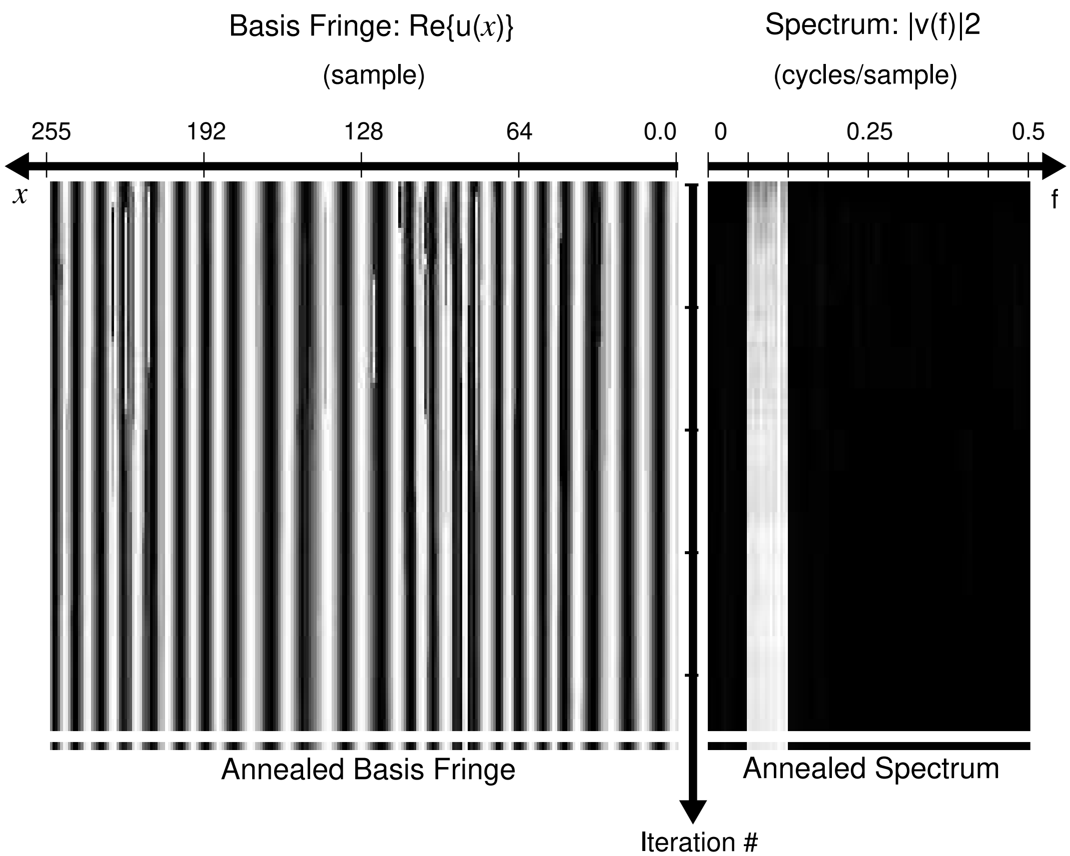
The Simulated Annealing Algorithm



*Rnd = a random number, (0.0,1.0]

This figure shows a schematic of the simulated annealing algorithm as implemented for the synthesis of basis fringes. Essentially, the fringe pattern gradually moves toward its targeted spectral constraints through a probabilistic decision to keep or reject randomly made changes. The annealing precedes until the error function *E* decreased to less than one quantization level per sample.

Simulated Annealing: An Example



This figure shows the process of simulated annealing applied to the synthesis of a basis fringe. At left is the real part of the spatial pattern. At right is the energy of the spectrum. The top line is the initial seed spatial pattern and its spectrum. Each line represents a change that was made and kept. The total number of changes attempted was 10000, and the number that were kept was 440. The bottom line is the annealed basis fringe and its spectrum. This basis fringe was targeted to have a uniform spectrum of 1.0 in the range [0.0625, 0.1250] and no spectral energy outside this range. The initial seed pattern was generated by the method of iterative constraints, which left some deviations from the targeted spectrum (even after 100 iterations). Notice the progression of the spectrum: it began with deviations in the form of dark regions within the region where it was targeted to be uniform. These variations virtually melted away as the annealing progressed. The error function decreased by roughly a factor of 10. Total time for 10000 iterations was two minutes.