Imaging-Consistent Warping and Super-Resolution

Ming-Chao Chiang

Department of Computer Science and Engineering

National Sun Yat-Sen University

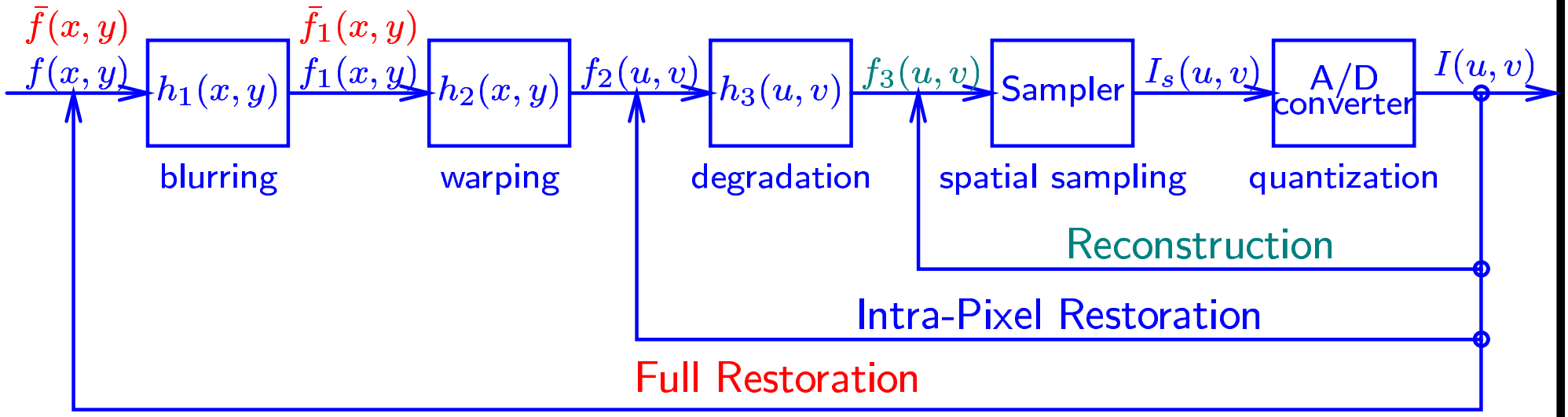
E-mail: mcchiang@mail.cse.nsysu.edu.tw

October 8, 2003

Outline

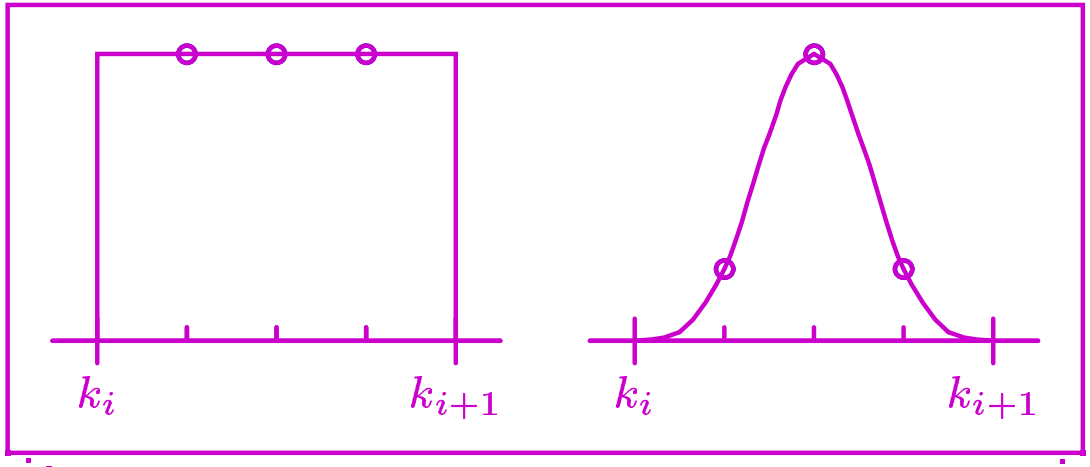
- Imaging-Consistent Algorithms
  - Reconstruction, Restoration, and Warping
- Super-Resolution Imaging
  - Image-Based Super-Resolution
  - Edge-Based Super-Resolution
- Quantitative Measurement of Super-Resolution Imaging
  - Using OCR
  - Using SLAM
- Conclusions

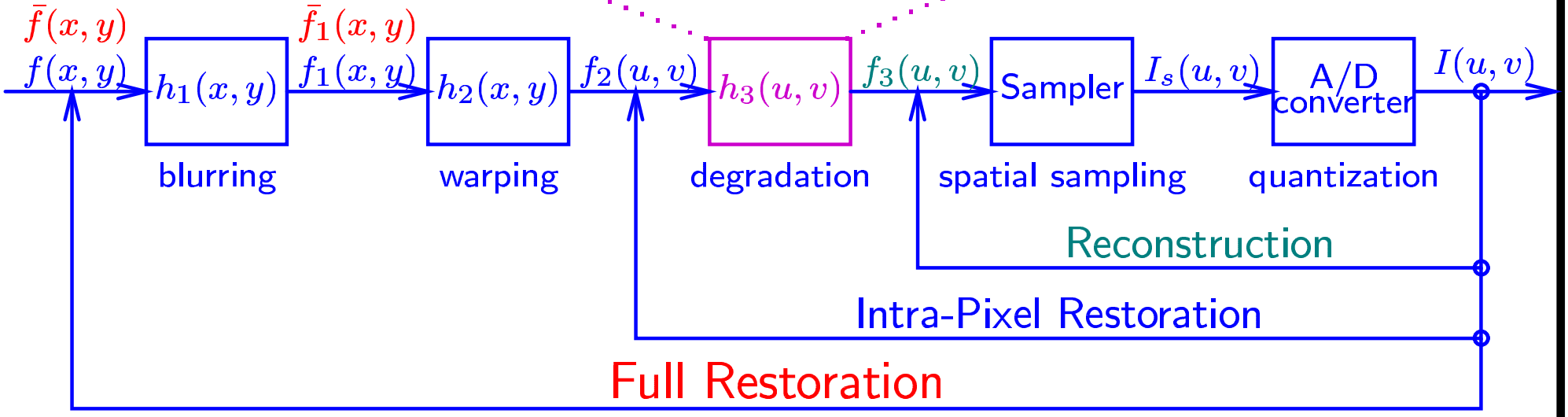
Relationship between Reconstruction, Restoration, and Super-Resolution



Reconstruction limits itself to the problem of deriving the continuous function f_3. Restoration attempts to estimate the original input function f_2, f_1, or f. The goal of super-resolution is to recover a discrete approximation, with resolution higher than I, to either $\bar{f}_1$ (plain super-resolution) or $\bar{f}$ (super-resolution with deblurring). Because super-resolution increases the signal-to-noise ratio, it significantly aids in the ill-condition problem of deblurring.

Our Sensor Model





Area Sampling

In this work, image values are treated as area samples generated by non-overlapping integrators.

Advantages

1. By coupling the PSF directly into the reconstruction and warping process, we satisfy a more intuitive fidelity measure of accuracy that is based on the physical limitations of the sensor.
2. The algorithms we derive with this approach satisfy an imaging-consistency property.

Traditionally

when used in warping, area sampling requires spatially varying filtering, i.e., non-linear, non-uniform sampling.

Imaging-Consistency

An algorithm is said to be imaging-consistent if it is the exact solution for some input from some space of allowable functions, which, according to the imaging model, would have generated the measured input.

- Imaging-consistent algroithms directly combine knowledge of image formation and sampling into the reconstruction and warping process.
- superior results are obtained by formulating reconstruction and warping as a two-stage process: image restoration followed by application of the PSF of the imaging sensor.

Imaging-Consistency

In [Boult and Wolberg 93], three algorithms are proposed.

1. A quadratic imaging-consistent algorithm assuming a Rect filter for the PSF.
2. A quadratic imaging-consistent algorithm assuming a cubic B-spline approximation to a Gaussian PSF.
3. An imaging-consistent algorithm with two cubic pieces per pixel assuming a Rect filter for the PSF.

In what follows, we'll only discuss a quadratic imaging-consistent algorithm assuming a Rect filter for the PSF.

Quadratic Imaging-Consistent Restoration Algorithm Assuming a Rect PSF

Given the values E_i at the pixel boundaries, an additional constraint, that the integral across the pixel must equal V_i, results in exactly three constraints:

$$q_i(-1/2) = E_i; \quad q_i(1/2) = E_{i+1}; \quad \int_{-1/2}^{1/2} q_i(x)\,dx = V_i.$$

The restored (deblurred) function (QRS) can be determined to be

$$q_i(x) = 3(E_i + E_{i+1} - 2V_i)x^2 - (E_i - E_{i+1})x - (E_i + E_{i+1} - 6V_i)/4$$

where $-1/2 \le x \le 1/2$.

Any method of reconstruction can be used to compute E_i. Using cubic convolution with the free parameter $A = 0$, we have

$$E_i = (V_{i-1} + V_i)/2 \quad \text{and} \quad E_{i+1} = (V_i + V_{i+1})/2.$$

Quadratic Imaging-Consistent Reconstruction Algorithm Assuming a Rect PSF

The reconstruction algorithm (QRR) can be defined by simply blurring the restored function (QRS) by the same PSF. The result is a cubic polynomial that spans from the center of one input pixel to the next.

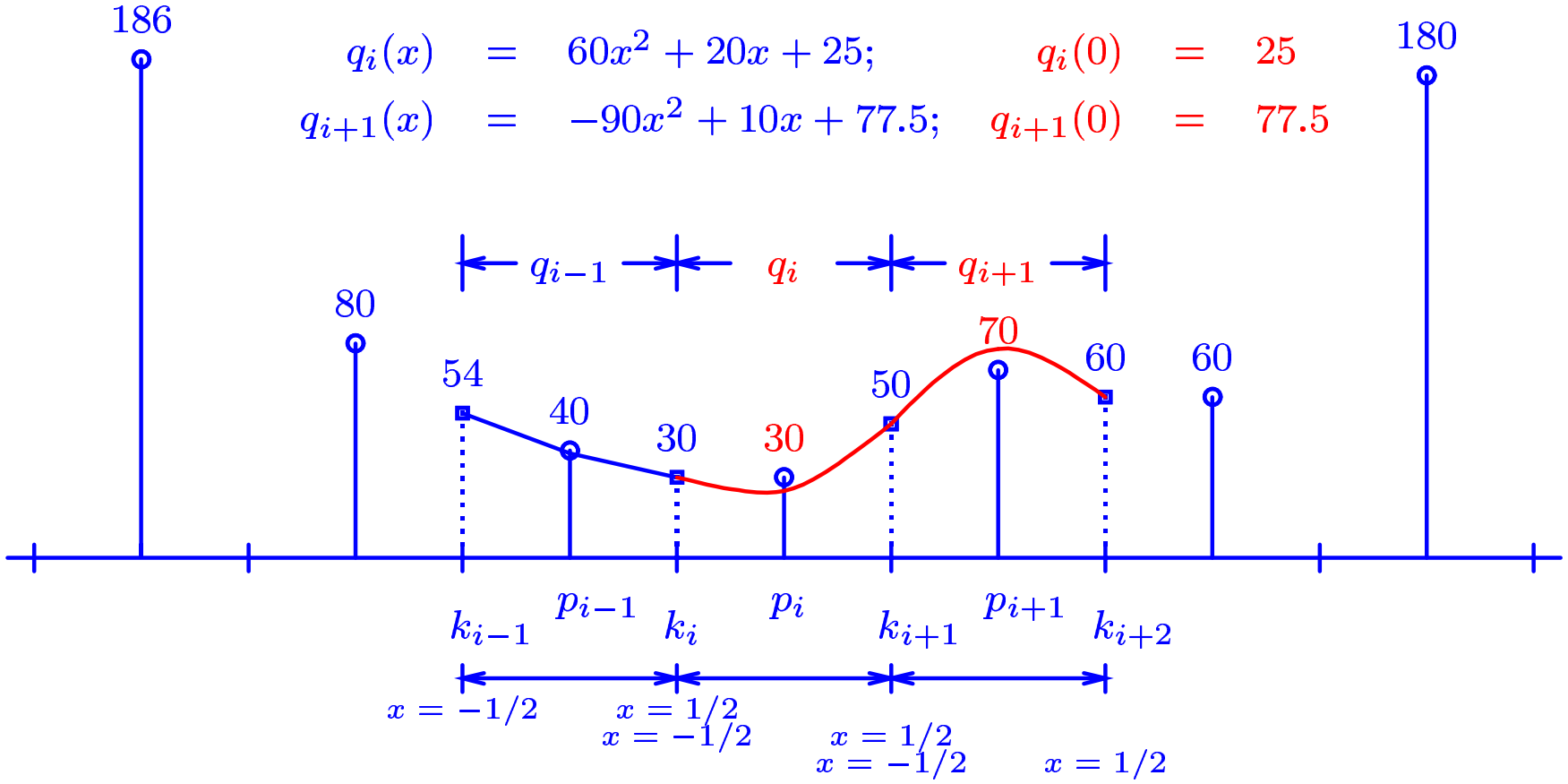
$$\begin{aligned} Q_i(x) &= \int_{x-1/2}^{1/2} q_i(z)\,dz + \int_{-1/2}^{x-1/2} q_{i+1}(z)\,dz \\ &= (E_{i+2} - E_i - 2(V_{i+1} - V_i))\,x^3 \\ &\quad + (2E_i - E_{i+1} - E_{i+2} + 3(V_{i+1} - V_i))\,x^2 \\ &\quad + (E_{i+1} - E_i)\,x + V_i \end{aligned}$$

where $0 \leq x \leq 1$.

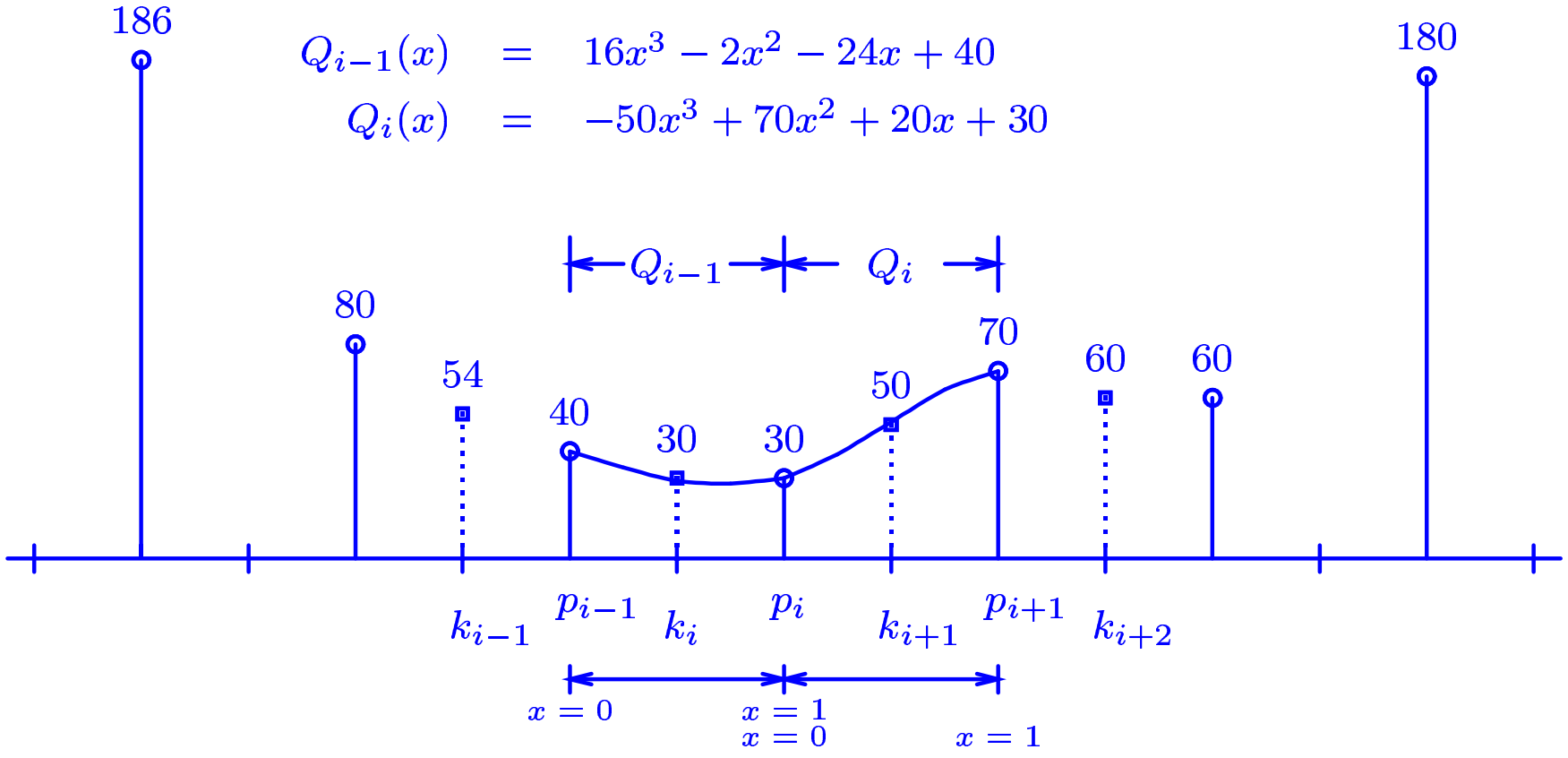
If linear interpolation is used to determine E_i, the resulting imaging-consistent reconstruction algorithm (QRR) is tantamount to cubic convolution with the "optimal" value of $A = -.5$.

QRS (A= -.5)

$$q_i(x) = 60x^2 + 20x + 25; \qquad q_i(0) = 25$$
$$q_{i+1}(x) = -90x^2 + 10x + 77.5; \qquad q_{i+1}(0) = 77.5$$



QRR (A= -.5)



Note that there is a half-pixel phase shift between the reconstruction and restoration.

Integrating Resamplers

The integrating resamplers provide an efficient method for image warping using the class of imaging-consistent reconstruction/restoration algorithms.

Whereas imaging-consistent reconstruction/restoration algorithms assume that the PSF is the same for both input and output; imaging-consistent warping algorithms go one step further, allowing

1. both input and output to have their own PSF, and
2. the PSF to vary its size for each output pixel.

Integrating Resamplers (cont'd)

The underlying idea of the integrating resampler can be found in the work of Fant who proposed a warping algorithm for the special case of piecewise constant reconstruction.

Our contribution is twofold:

1. the generalization to deal with more advanced reconstruction algorithms, and
2. providing for a modeling of real lens effects by using a modeling of real warps that affects the image radiance.

Integrating Resamplers (cont'd)

- The goal of Fant's original work was to warp images for graphic (or visual) effects, i.e., to affect geometry without disturbing the intensities.

  Thus, if a constant image was stretched to twice its normal width, it would just change shape but retain the same intensities.

- Our goal is to the modeling of lens characteristics; we realized that we do not want to do this normalization.

  Thus, if a lens was placed into an imaging system so as to double the width of the image on the sensor plane, then the value measured would be halved.

Integrating Resamplers (cont'd)

By coupling the PSF of the imaging system directly into the integrating resampler, we can better approximate the warping characteristics of real sensors, which also significantly improve the accuracy of the warped intensity values.

When we are resampling the image and warping its geometry in a non-linear manner, the integrating resampler

1. allows us to efficiently do both pre-filtering and post-filtering. Because we have a functional form for the input, no spatially-varying filtering is needed, as would be the case if direct inverse mapping were done.
2. also handles antialiasing of partial pixels in a straightforward manner.

Integrating Resamplers (cont'd)

Pad the input; compute k_l, k_r, and i_l, the indices to the leftmost and rightmost output pixels and the index to the leftmost input pixel that contributes to the output; and compute the linear approximation to the location of $m^{-1}(j)$, for $j = k_l, \ldots, k_r+1$.

```
normalizingfactor = q_{k_l+1} - q_{k_l};        /*  set up for normalization */
q_{k_l} = MAX(q_{k_l}, 0);                      /*  ensure that q_{k_l} is nonnegative */
inseg = 1.0 - FRACTION(q_{k_l});                /*  fraction of input pixel left to be consumed */
outseg = q_{k_l+1} - q_{k_l};                   /*  #input pixels mapped onto one output pixel */
acc = 0.0;                                      /*  reset accumulator for next output pixel */
for (j = 0; j < q_{k_l}; j++) out[j++] = 0;     /*  zero out the garbage at left end */
for (i = i_l, j = k_l; j <= k_r; ) {            /*  while there is output to produce */
  Use the current pixel (in[i]) and its neighbors to update R(), the integral of the restoration r().
  leftpos = 1.0 - inseg;                        /*  get left endpoint for integration */
  if (inseg < outseg) {                         /*  if we will consume input pixel first */
    acc += R(1) - R(leftpos);                   /*  add integral to end of output pixel */
    i++;                                        /*  index into next input pixel */
    if (i == n) {                               /*  check end condition */
      if (normalize) acc /= normalizingfactor;  /*  normalize the output, if appropriate */
      out[j] = acc;                             /*  init output */
      break;                                    /*  exit from the loop */
    }
    outseg -= inseg;                            /*  inseg portion has been filled */
    inseg = 1.0;                                /*  new input pixel will be available */
  }
  else {                                        /*  Else we will produce output pixel first */
```

Integrating Resamplers (cont'd)

```
    acc += R(leftpos + outseg) − R(leftpos);   /*  add integral to end of output pixel */
    if (normalize) acc /= normalizingfactor;   /*  normalize the output, if appropriate */
    out[j] = acc;                              /*  init output */
    j++;                                       /*  index into next output pixel */
    acc = 0.0;                                 /*  reset accumulator for next output pixel */
    inseg −= outseg;                           /*  outseg portion of input has been used */
    outseg = q_{j+1} − q_j;                    /*  new output size */
    normalizingfactor = outseg;                /*  need for normalization */
  }
}
for (j = k_r + 1; j < k; j++) out[j++] = 0;    /*  zero out the garbage at right end */
```

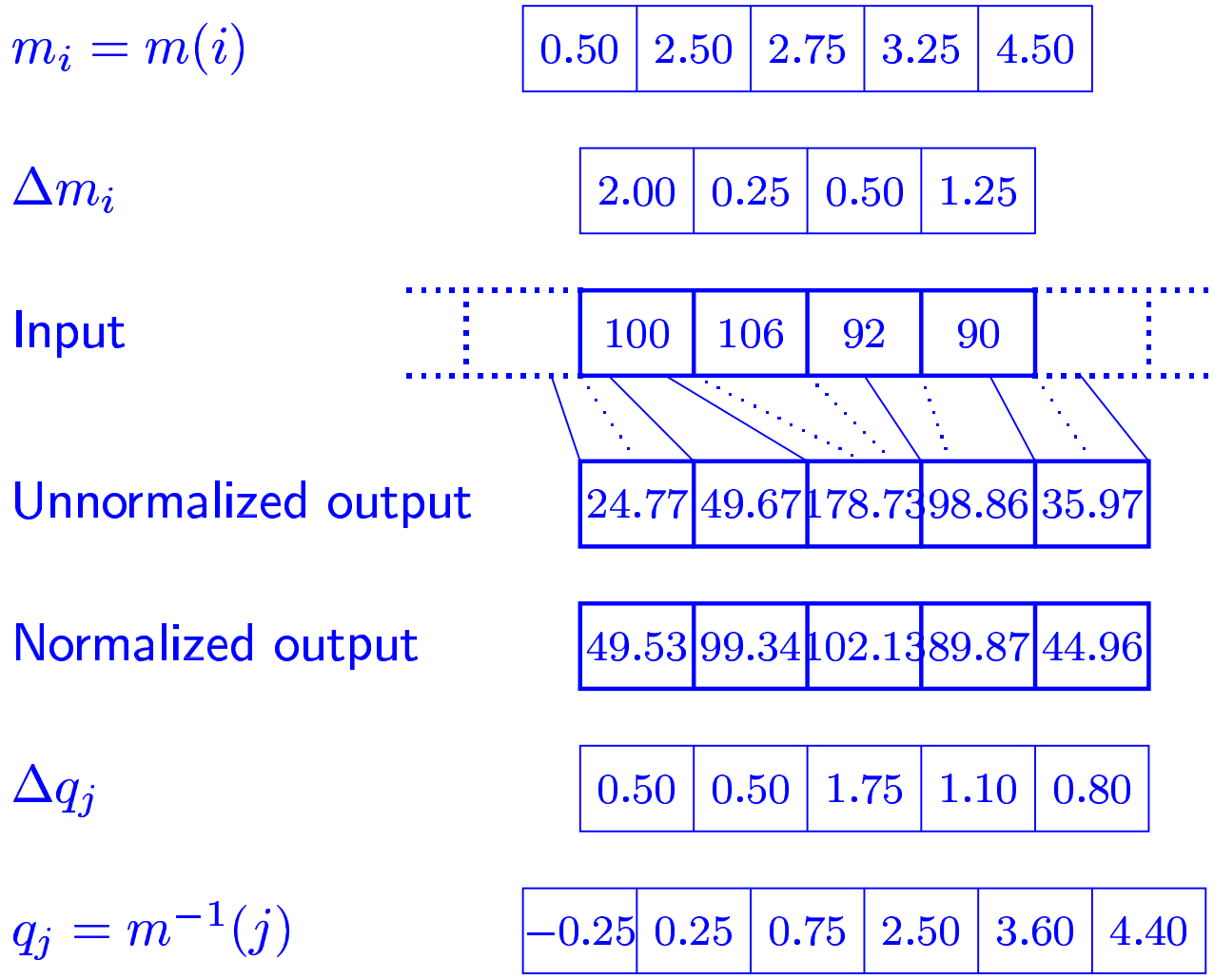
Linear approximation to the location of $m^{-1}(j)$

Assume n input pixels are being mapped into k output pixels according to the mapping function $m(t)$. Let m_i be the mapped location of pixel i, for $i{=}0, \ldots, n$. The linear approximation to the location of $m^{-1}(j)$, q_j, $j{=}0, \ldots, k$ can be computed as follows:

```
for (i = j = 0; j ≤ k; j++) {
  while (i < n − 1 && m_{i+1} < j) i++;
  q_j = i + (j − m_i)/(m_{i+1} − m_i);
}
```

Note that we are assuming that the mapping function is strictly increasing to avoid fold-over problems. See [Wolberg89] for an approach to modeling fold-over.

Integrating resampling example assuming QRR -.5



Super-Resolution

- The central idea of super-resolution is to combine information from images to produce an image that has higher "resolution" than the original image.

  Even if the final sampling rate is exactly the same as the original, we can STILL do super-resolution. In this case, it provides a sharper image of the original size.

- Just increasing the sampling rate is NOT super-resolution because it does not provide additional information.

  For instance, scaling up an image by a factor of 2 is not super-resolution no matter what interpolation techniques are used.

What Is Super-Resolution?

Super-resolution is about increasing SNR. It does this by combining data. It

- "averages" to reduce random fluctuation
- reduces aliasing errors by using different views
- provides better edge definition by its robust estimation

Restoration is noise sensitive. Super-resolution makes it more reliable.

Problem with traditional multi-image restoration:

- Need solve large linear systems (very slow)
- Depend heavily on correct lens/noise models

Earlier Research on Super-Resolution

- Huang and Tsai solved the problem in the frequency domain, but they disregarded sensor blurring. Furthermore, they assumed only inter-frame translations.
- Gross solved the problem by first merging the low-resolution images by interpolation and then obtaining the high-resolution image by deblurring the merged image. Again, only inter-frame translations were considered.
- Peleg and Keren solved the problem by first estimating an initial guess of the high-resolution image, then simulating the imaging process and minimizing the difference between the observed and simulated low-resolution images.
- Irani and Peleg used a back-projection method similar to that used in tomography to minimize the same difference.
- Bascle et al. used the same back-projection method to minimize the difference between the predicted and actual images except that a simple motion blur model was included.

Problem with Previous Work in Super-Resolution

All previous work ignores the impact of image warping techniques.

Previous work uses the PSF to simulate the imaging process, which is computationally expensive.

We show that

1. Warping techniques may have a strong impact on the quality of image resolution enhancement.
2. By coupling the PSF of the imaging system directly into the integrating resampler, we can better approximate the warping characteristics of real sensors and significantly improve the quality of super-resolution imaging.

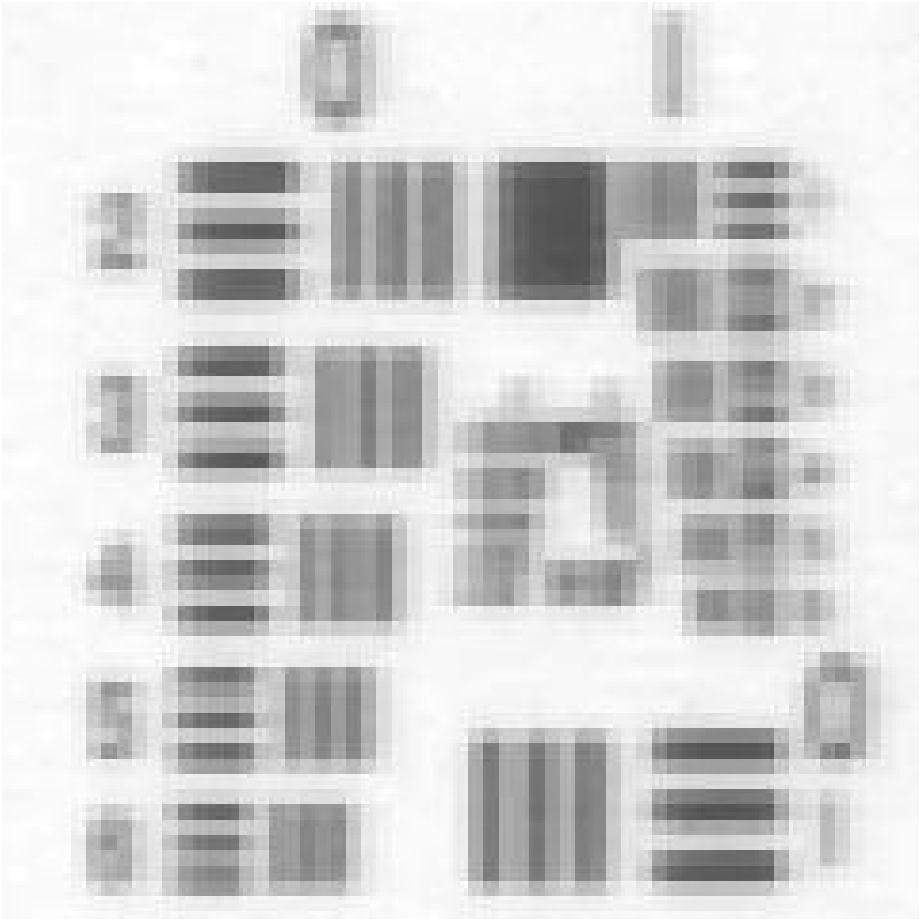
The Image-Based Super-Resolution Algorithm

Step 1 Choose one of the images as the reference image, and compute the motion field between all the images and the reference image.

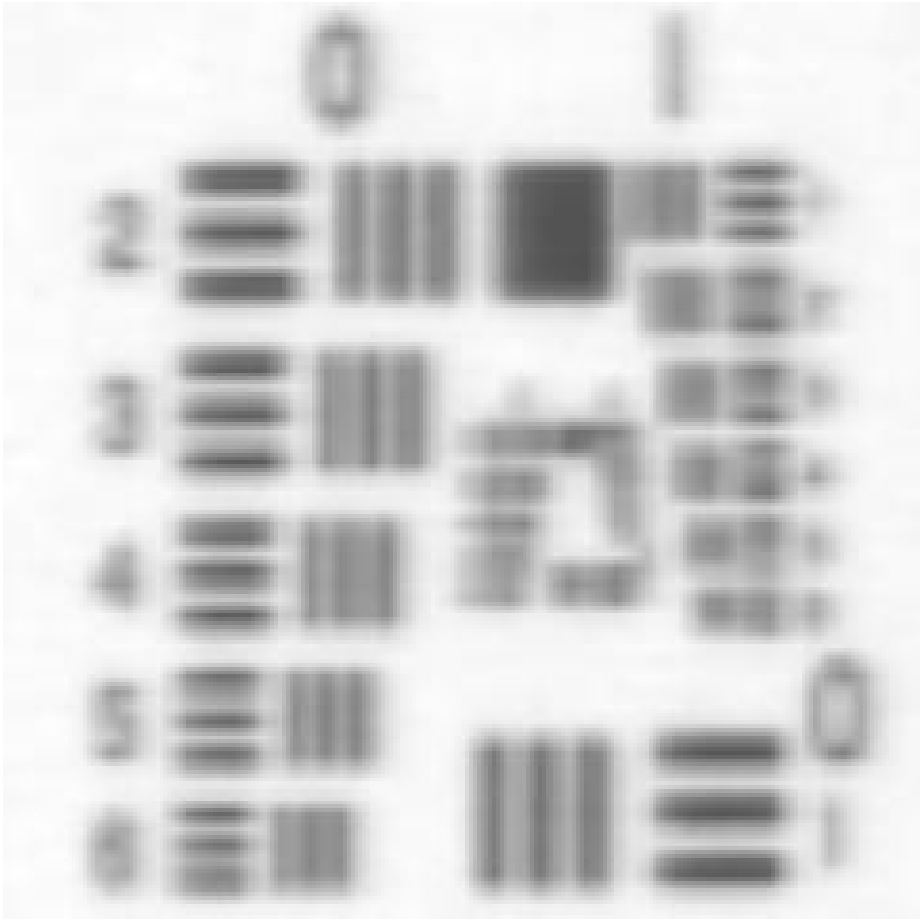
Step 2 Scale up the reference image, and then scale/warp all the images to the reference image based on the motion field computed in the previous step and the chosen scale.

Step 3 Obtain a super-resolution image by fusing all the images together. We considered both averaging and median "fusion".

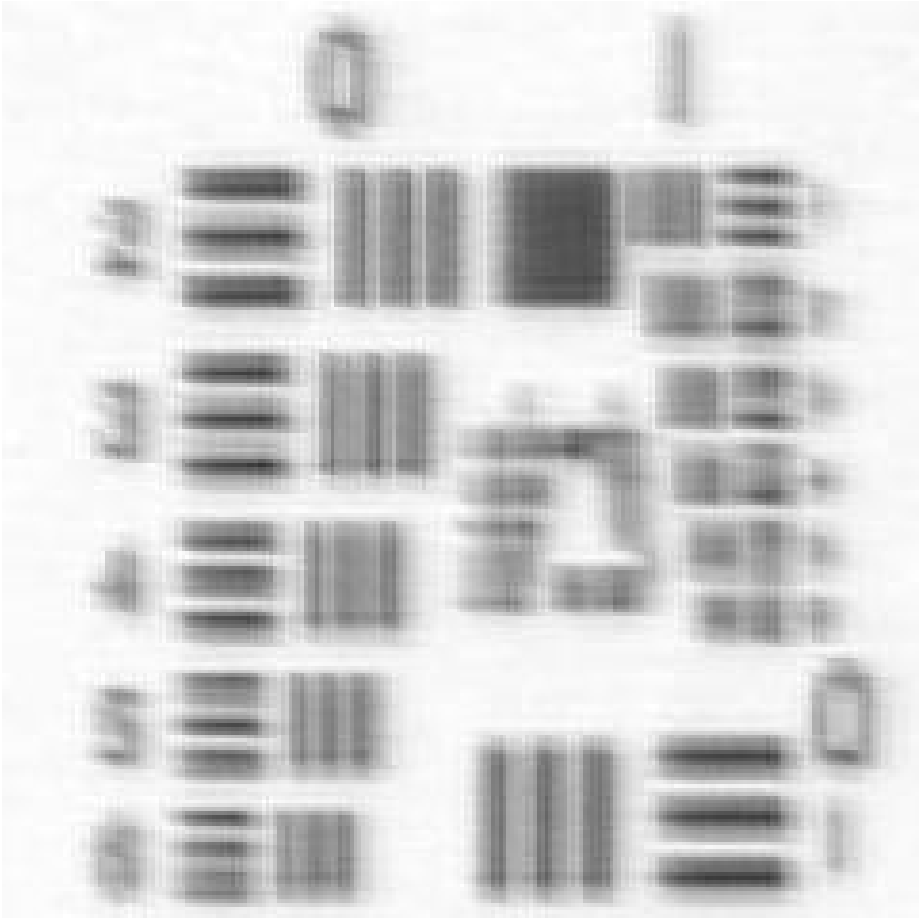
Step 4 Optional deblurring stage.



4X Pixel Replication

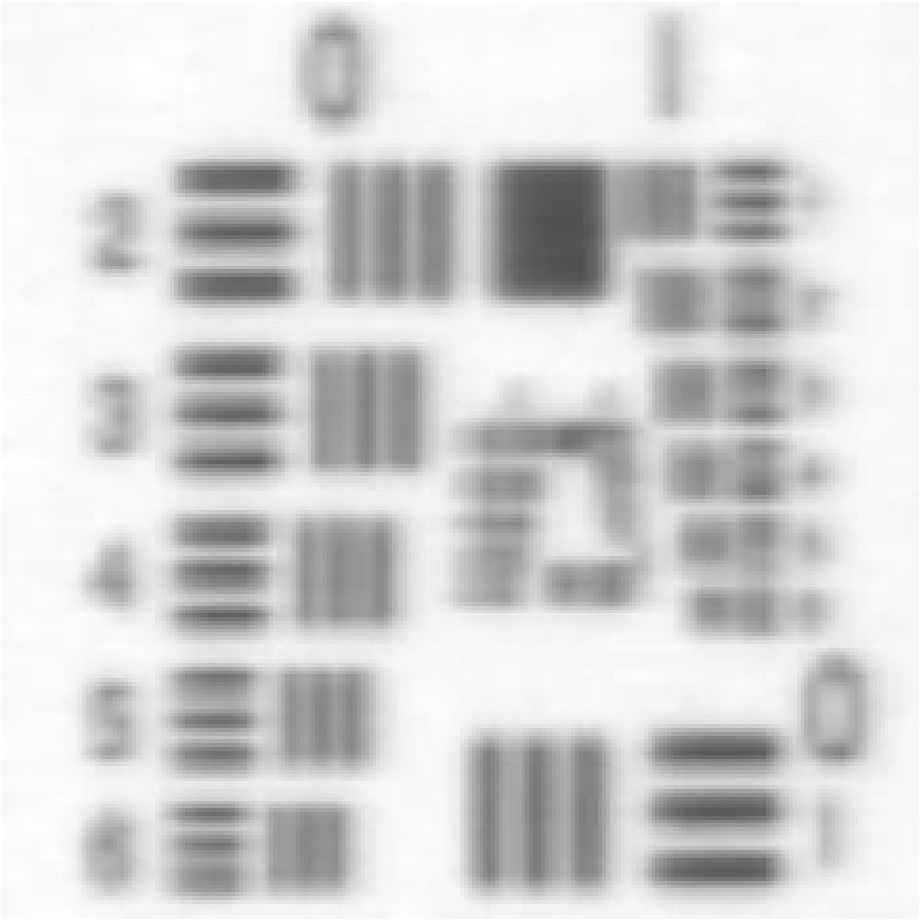


4X Bilinear Resampling

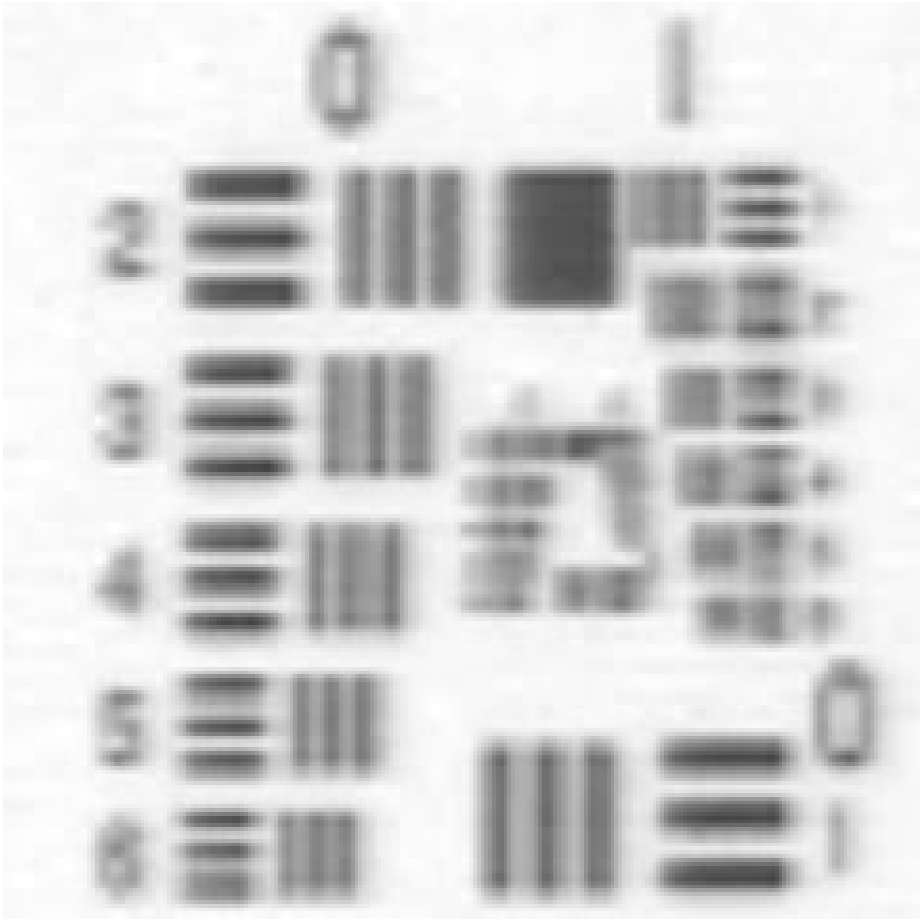


Irani and Peleg: Best Prior Work
(Back Projection)

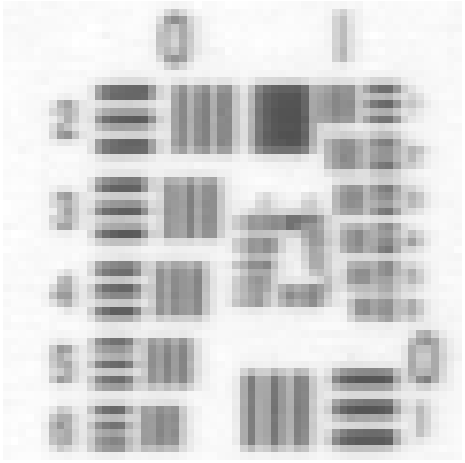
New Algorithm
(SR QRW w/ deblurring)



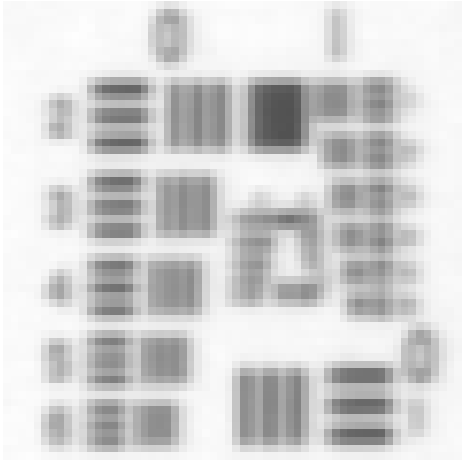
SR Bilinear before deblurring



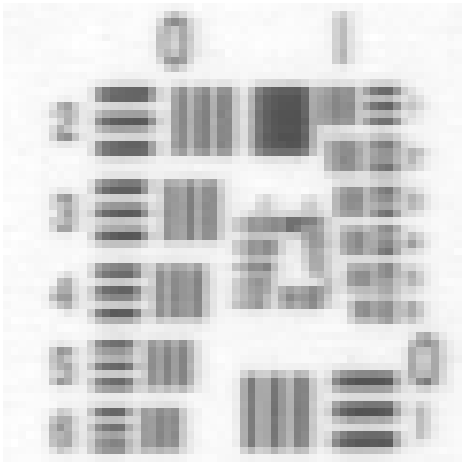
SR QRW before deblurring



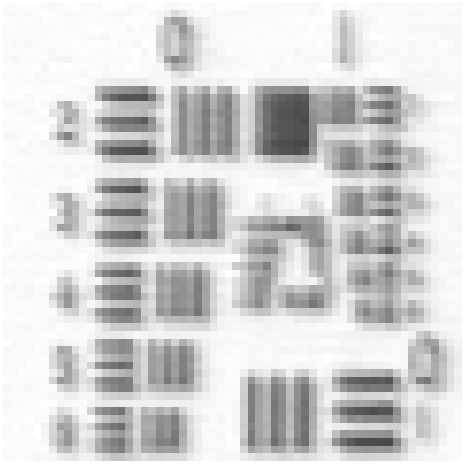
Original



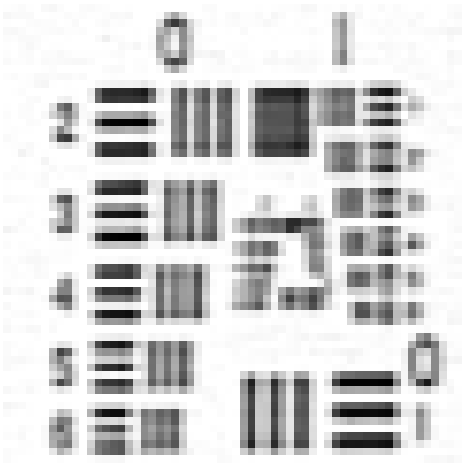
SR Bilinear



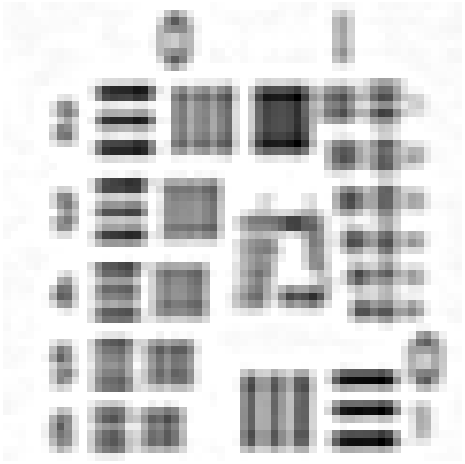
SR QRW



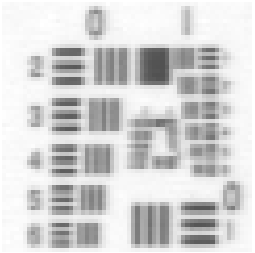
Back Projection



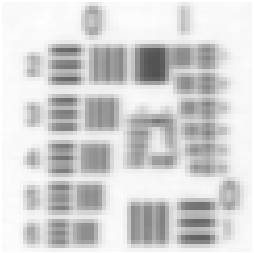
SR QRW Deblur



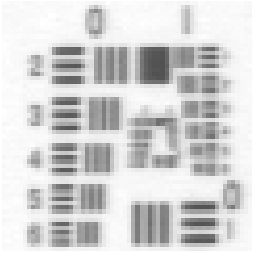
Original Deblur



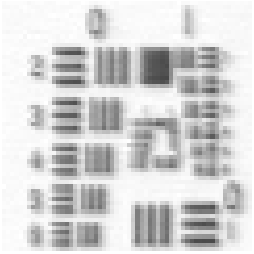
Original



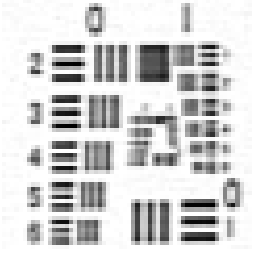
SR Bilinear



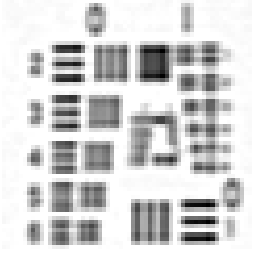
SR QRW



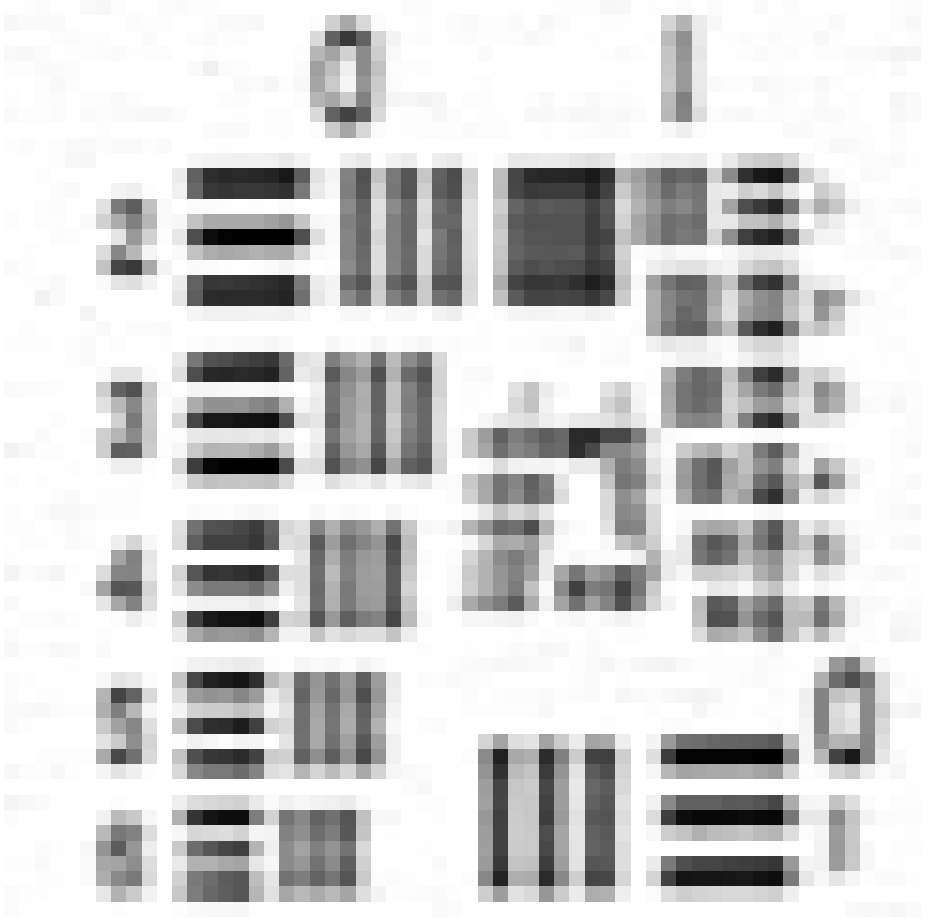
Back Projection



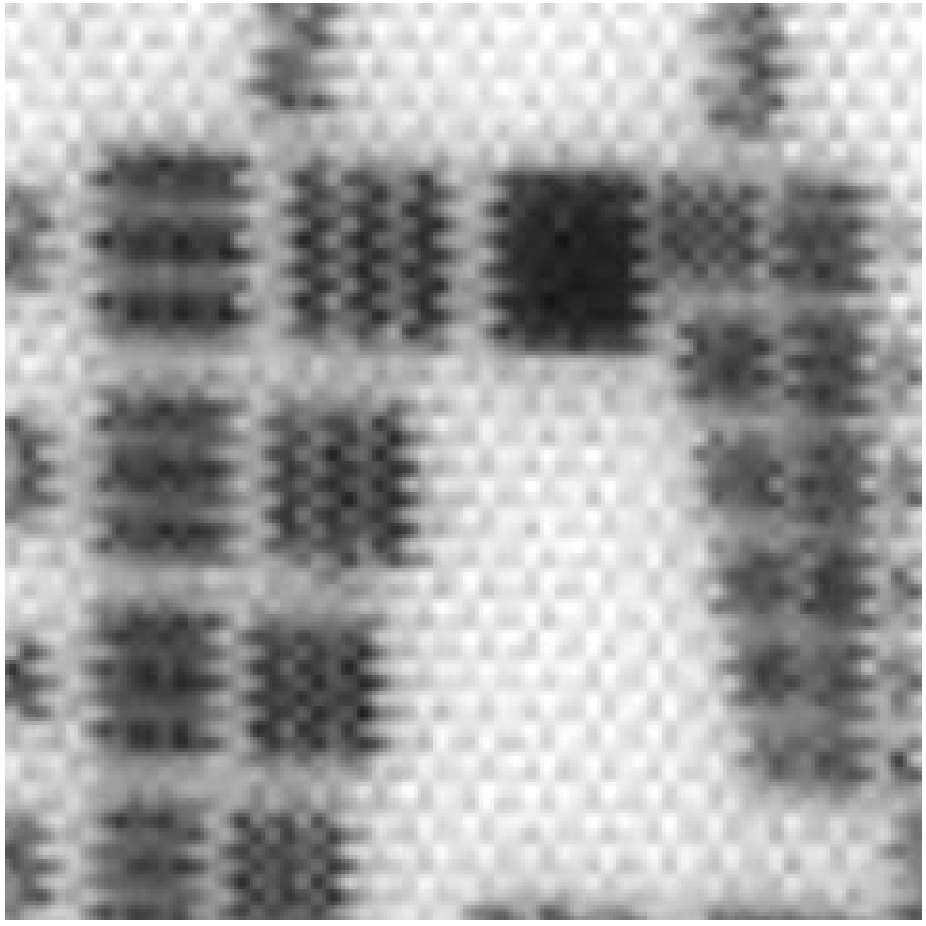
SR QRW Deblur



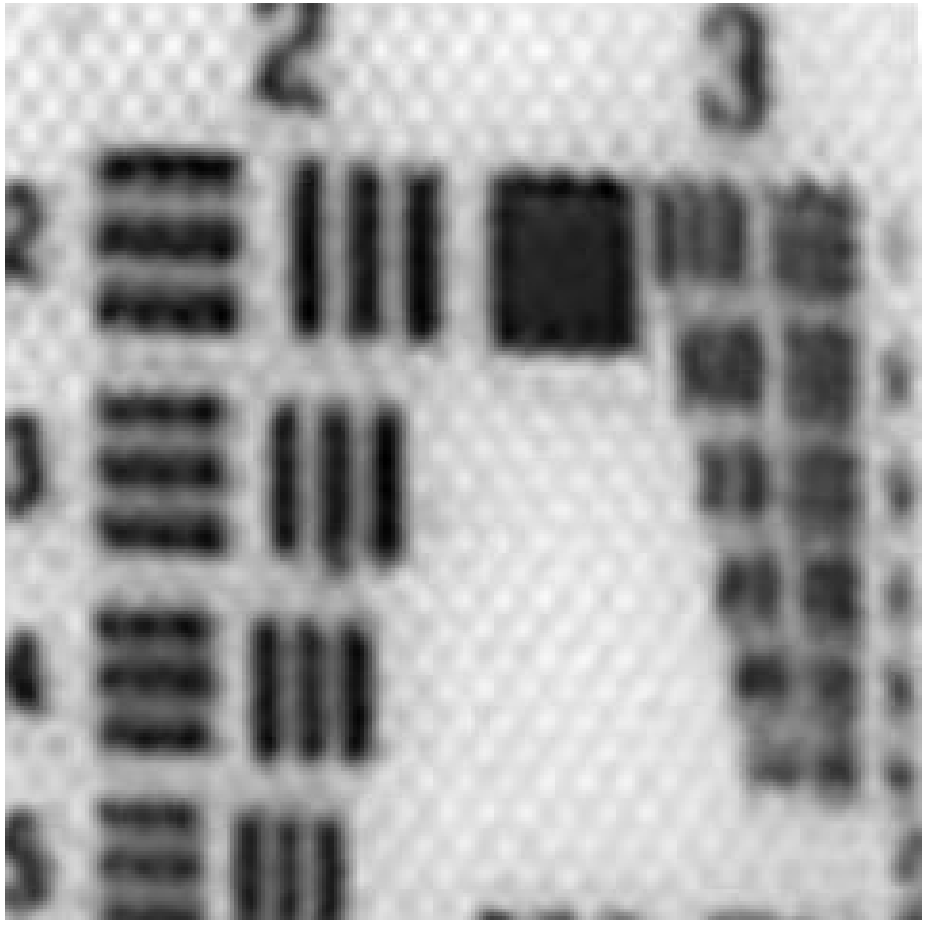
Original Deblur



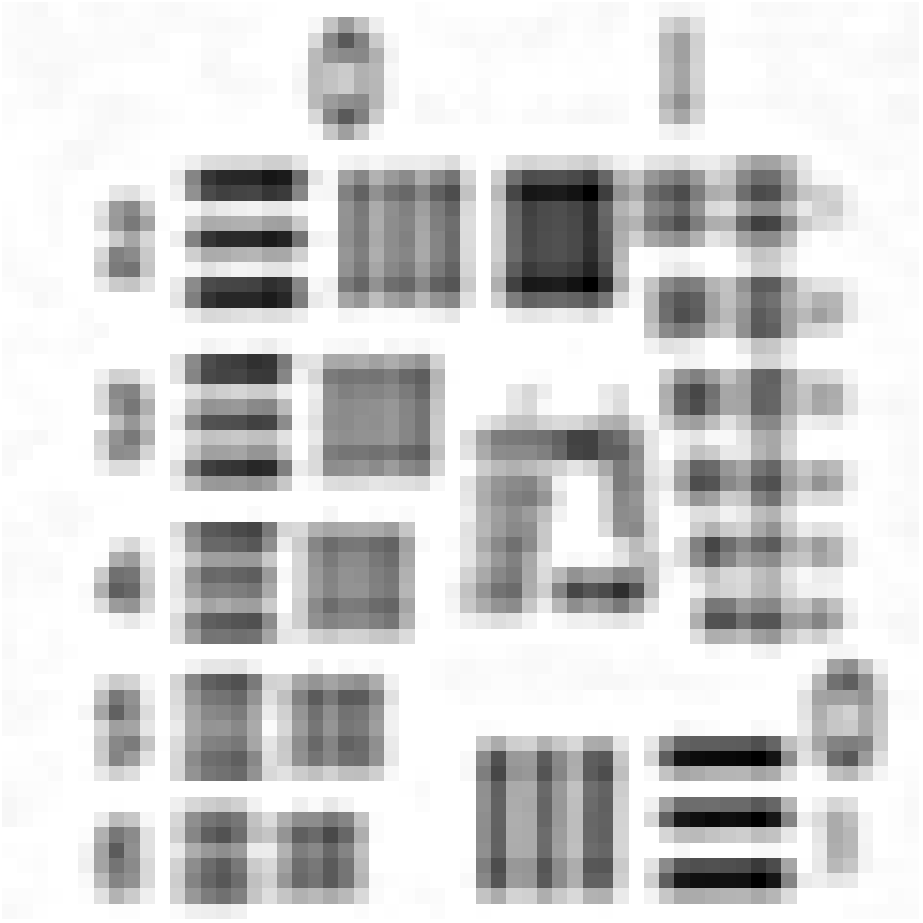
SR QRW Deblur 4X Pixel Replication



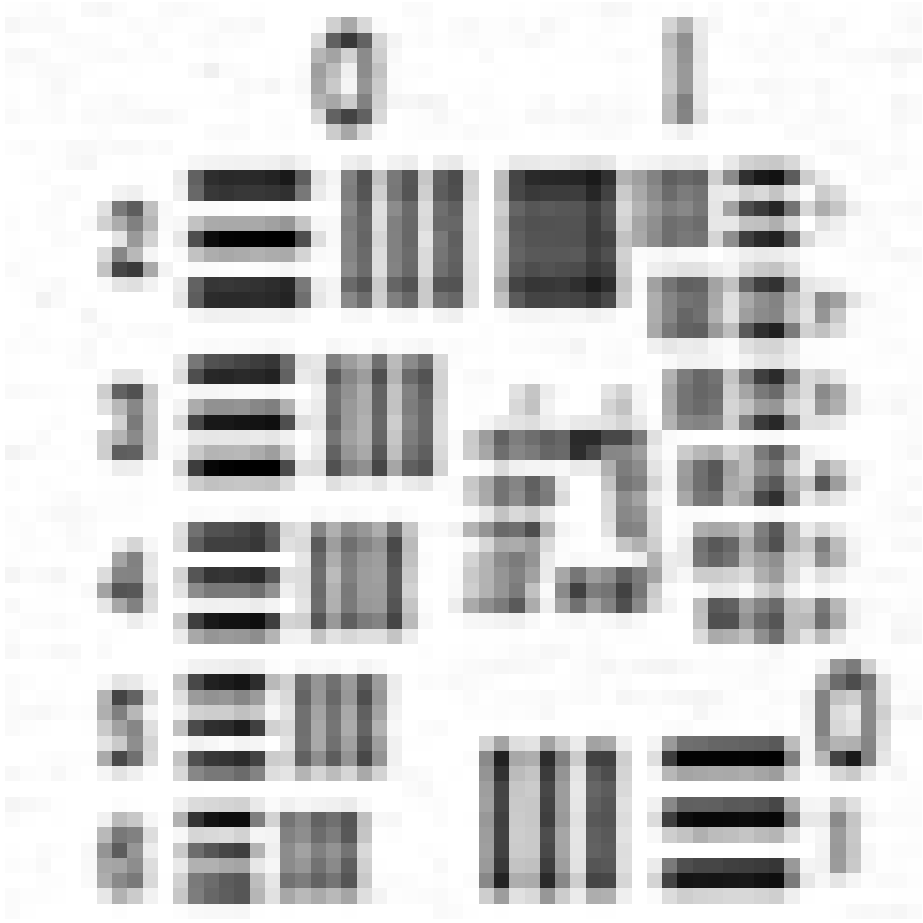
4X Pixel Replication



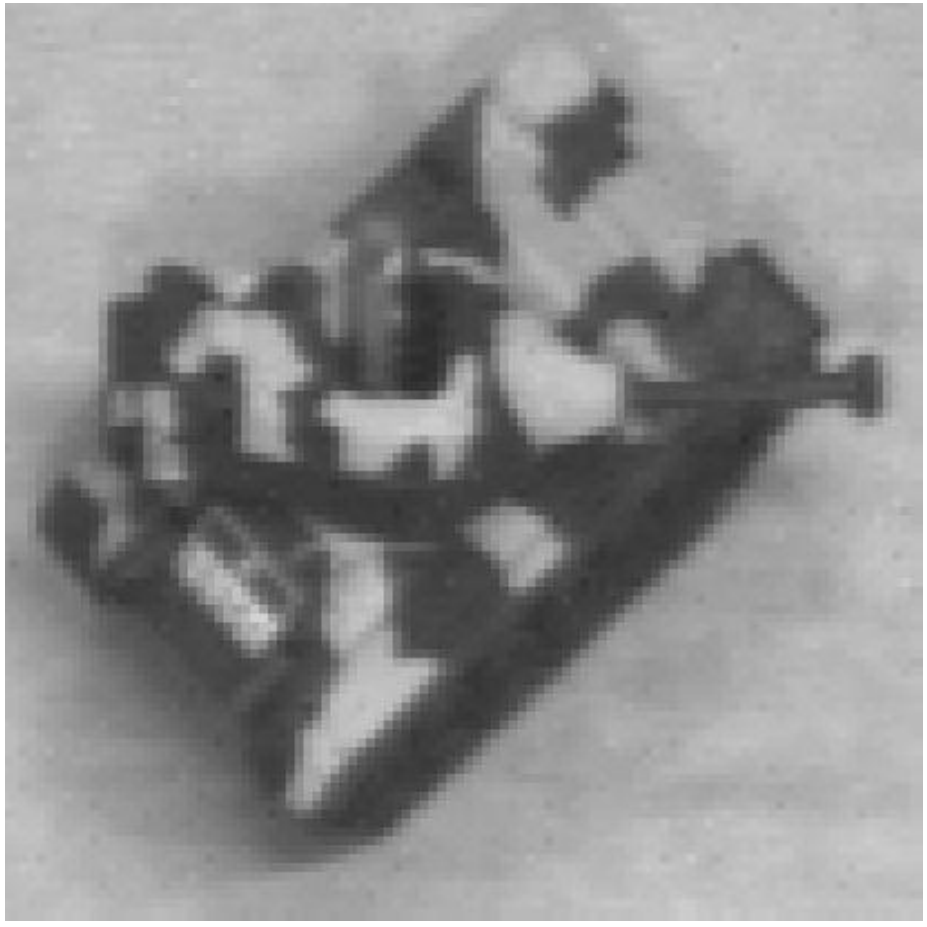
SR QRW w/ deblurring



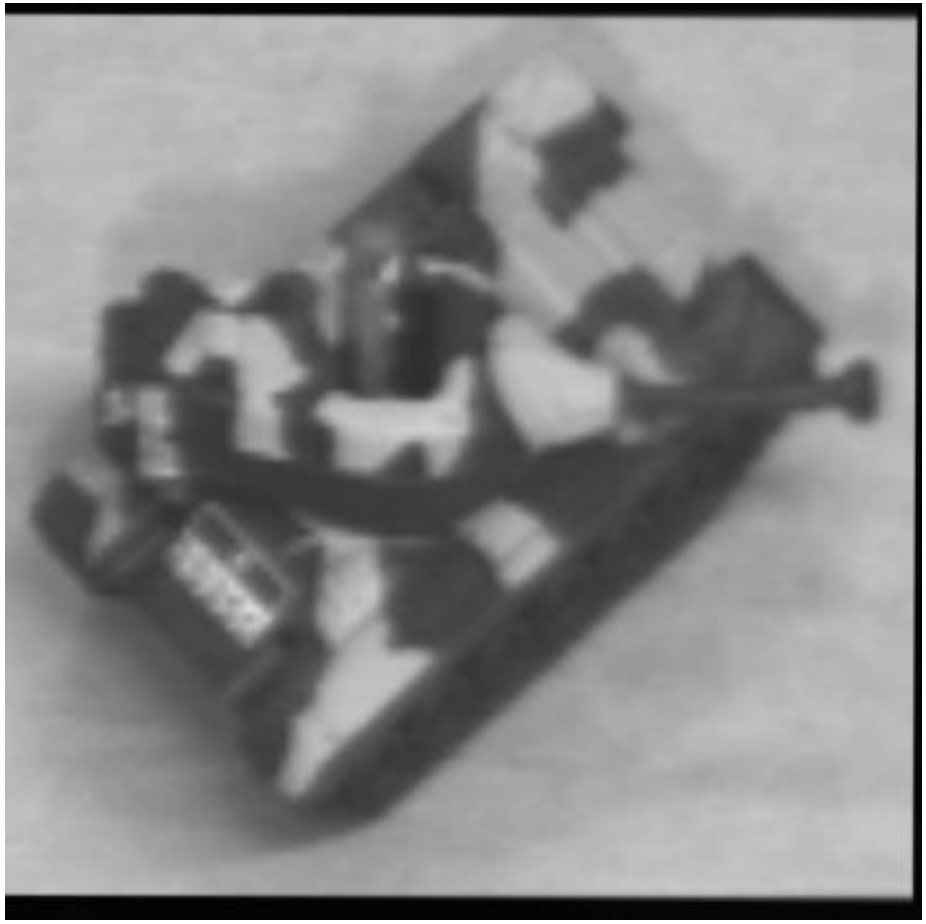
Original Deblur 4X Pixel Replication



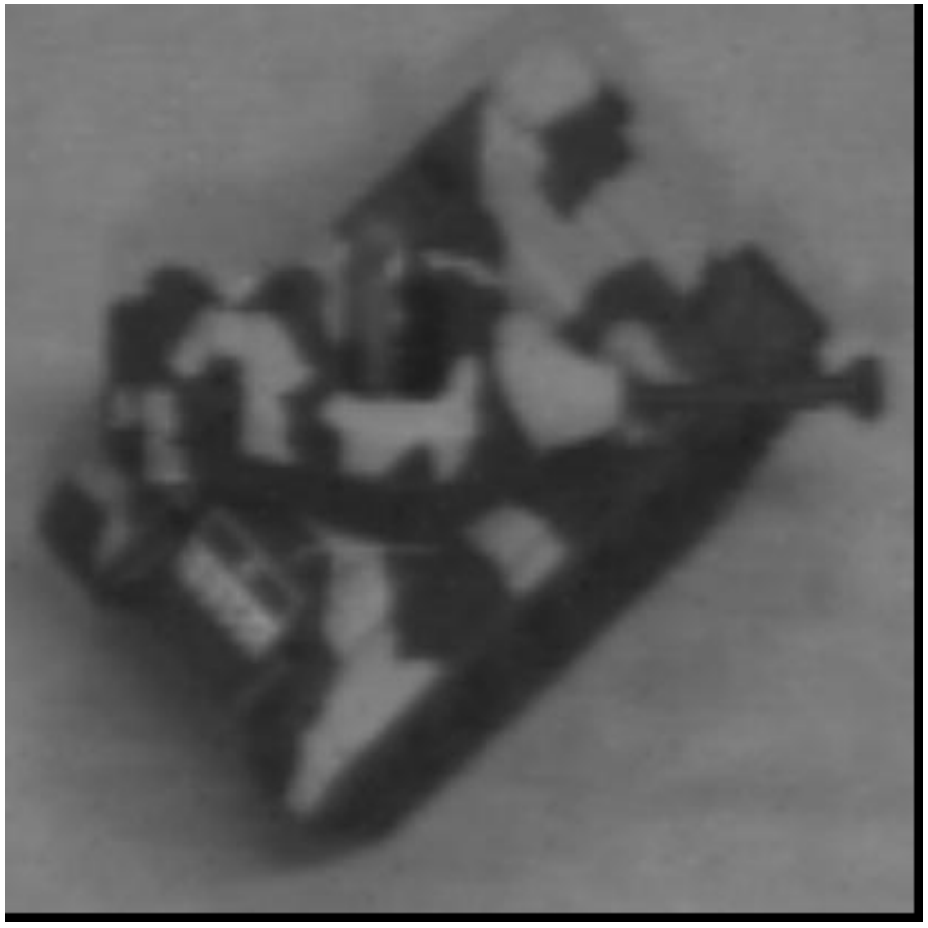
SR QRW Deblur 4X Pixel Replication



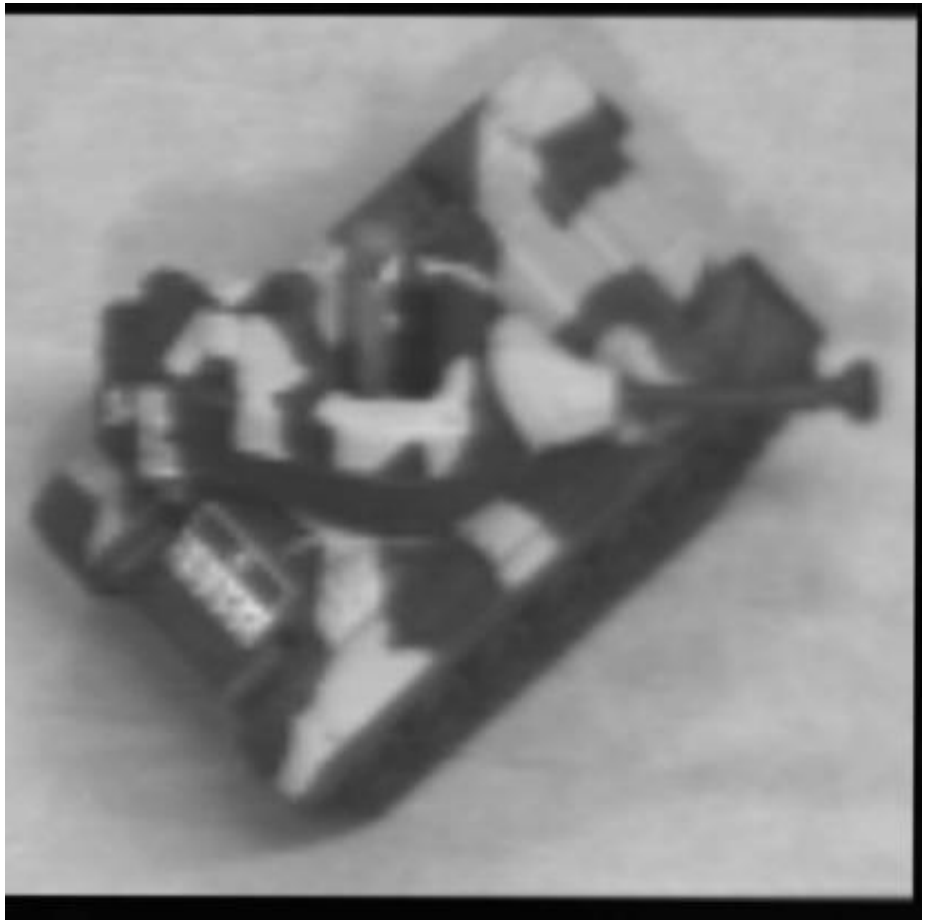
4X Pixel Replication



4X SR QRW w/ deblurring



4X Bilinear Resampling



4X SR QRW w/ deblurring

Running Times of Our First Experiment

Running times in seconds	QRW		Back-projection	
	SPARC	Pentium	SPARC	Pentium
Warping	1.94	3.52	-	-
Fusion	0.04	0.25	-	-
Deblurring	0.81	1.31	-	-
Total	2.79	5.08	12.33	24.80

The running times are measured on a Sun 143MHz Ultra SPARC running Solaris 2.5 and a 120MHz Pentium running Linux 2.0.12. Running times required by both methods such as computation of the motion field are not included.

- For precision reasons, all the operations are done in double-precision floating point.
- Irani's back-projection method takes time roughly proportional to the number of iterations as well as the size of the PSF.

- Our experiments show that
  - Each iteration of Irani's back-projection method takes only about two thirds the running times of our method; however, at least two iterations are required to determine when to stop iterating. In general, more than three iterations are often required to minimize the sum-of-square difference.
  - Our method is often more than two or three times faster than Irani's back-projection method.

Problem with Image-Based Super-Resolution

All previous super-resolution algorithms presume that the images were taken under the same illumination conditions and use the intensity information provided by the image sequence to construct the super-resolution image.

What happens if lighting changes?

For example, when imaging from different viewpoints, over long temporal spans, or imaging scenes with moving 3D objects, the image intensities naturally vary.

So, how do we eliminate this undesirable effect?

Use edges to avoid lighting variations, but how to use it in super-resolution.

Edge-Based Super-Resolution

The edge-based approach, based on edge models and a local blur estimate, circumvents the difficulties caused by lighting variations.

Rather than using the intensity information provided by all the images to construct the super-resolution image, we may choose one, and only one, of the images from the image sequence and then fuse together all the edges from the other images.

However, to fuse all the edges together requires that

1. the edges be detected and then warped, and
2. the reference image be reestimated and scaled up based on the edge models and local blur estimation.

- Therefore, we generalize the idea of the imaging-consistent algorithms to do reconstruction using edges and intensities.
- We introduce new algorithms for edge localization and local blur estimation algorithm, based on QRR.
- The idea of edge-based super-resolution is to fuse all the edge models together and use intensity levels from a single image.

Edge Localization

Typically, edge detection involves the estimation of first or second derivatives of the luminance function, followed by selection of extrema or zero crossing.

We propose a close form solution based on QRR.

No convolution step

Given $Q(x)$ (i.e., $Q(x) = ax^3 + bx^2 + cx + d$), our edge detection algorithm is as follows:

1. Find the slope function $S(x)$ that gives all the slopes of $Q(x)$.

$$S(x) = Q'(x) = 3ax^2 + 2bx + c$$

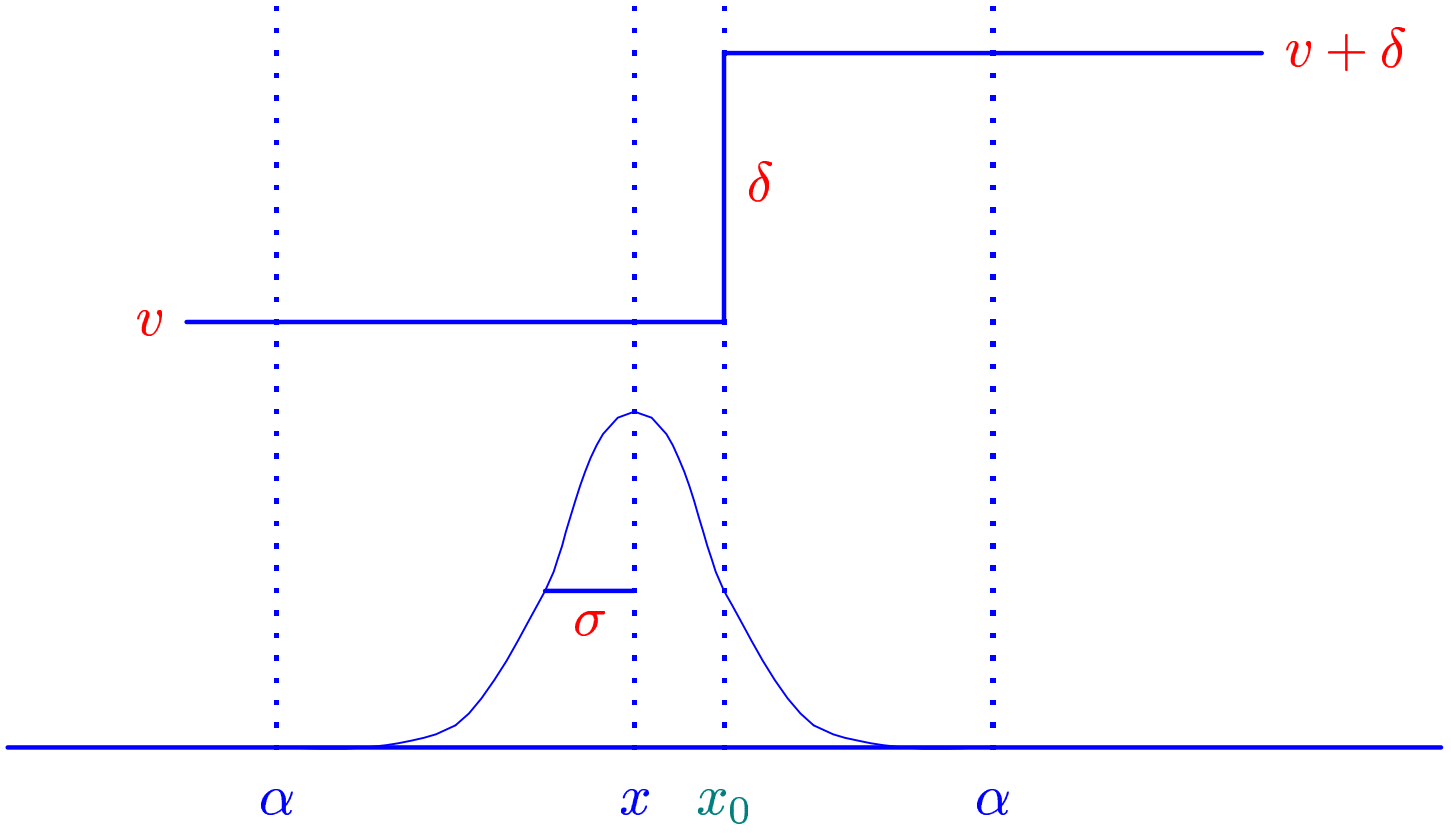
2. Find edge position with sub-pixel accuracy.

   Letting $S'(x) = 6ax + 2b = 0$, the position of the step edge can be found located at $x_0 = -b/3a$, and the slope is given by $S(x_0) = -b^2/3a + c$.

3. Find edge direction and magnitude.

$$E_d = \begin{bmatrix} s_x \\ s_y \end{bmatrix} \quad \text{and} \quad E_m = \sqrt{s_x^2 + s_y^2}.$$

Edge Model



v, δ, and σ are the three unknowns. α is a predefined nonnegative constant. x_0 is the location of the step edge.

- We model an edge as a step function $v + \delta u(x)$ where v is the unknown intensity value and δ is the unknown amplitude of the edge.

- The focal blur of this edge is modeled by a "truncated" Gaussian blur kernel

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

  where σ is the unknown standard deviation.

- There are total three unknowns—v, δ, and σ.

The complete edge model is thus given by

$$B(x) = \begin{cases} \int_{x-\alpha}^{x+\alpha} vG(x-z)\,dz, & x < x_0 - \alpha \\ \int_{x-\alpha}^{x_0} vG(x-z)\,dz + \int_{x_0}^{x+\alpha} (v+\delta)G(x-z)\,dz, & x_0 - \alpha \le x \le x_0 + \alpha \\ \int_{x-\alpha}^{x+\alpha} (v+\delta)G(x-z)\,dz, & x > x_0 + \alpha \end{cases}$$

After expanding and simplifying, we have

$$B(x) = \begin{cases} v\,\mathsf{erf}\left(\dfrac{\alpha}{\sqrt{2}\sigma}\right), & x < x_0 - \alpha \\ (v+\dfrac{\delta}{2})\,\mathsf{erf}\left(\dfrac{\alpha}{\sqrt{2}\sigma}\right) + \dfrac{\delta}{2}\,\mathsf{erf}\left(\dfrac{x-x_0}{\sqrt{2}\sigma}\right), & x_0 - \alpha \le x \le x_0 + \alpha \\ (v+\delta)\,\mathsf{erf}\left(\dfrac{\alpha}{\sqrt{2}\sigma}\right), & x > x_0 + \alpha \end{cases}$$

Local Blur Estimation

Since there are three unknowns, we need three constraints to solve the system of equations.

$$\begin{aligned} Q(x) &= B(x) \\ Q'(x) &= B'(x) \\ Q'''(x) &= B'''(x) \end{aligned}$$

To ensure that the general edge model gives exactly the same edge model located at x_0, we use the third derivative instead of the second derivative for solving the system of equations.

Solving the system of equations for the three unknowns v, δ, and σ, and noting that the value of σ must be positive, we have the complete edge model.

$$\begin{aligned}
\sigma &= +\sqrt{\frac{r+\sqrt{s}}{2}} \\
\delta &= \left(\sqrt{2\pi}\sigma Q'(x)\right) / \exp\left(-\frac{(x-x_0)^2}{2\sigma^2}\right) \\
v &= \left(Q(x) - \frac{\delta}{2}\,\mathrm{erf}\left(\frac{x-x_0}{\sqrt{2}\sigma}\right)\right) / \mathrm{erf}\left(\frac{\alpha}{\sqrt{2}\sigma}\right) - \frac{\delta}{2}
\end{aligned}$$

where $r = -Q'(x)/Q'''(x)$ and $s = r^2 - 4r(x-x_0)^2$.

- Since the value of σ must be positive, it follows that σ has a solution if and only if $s > 0$ and either $r + \sqrt{s} > 0$ or $r - \sqrt{s} > 0$; σ has two solutions if $s > 0$ and both $r + \sqrt{s} > 0$ and $r - \sqrt{s} > 0$.
- However, to ensure that the general edge model gives exactly the same edge model located at x_0, only $r + \sqrt{s} > 0$ yields a valid solution.
- It follows that the solution of σ is unique and is given by

$$\sigma = \sqrt{\frac{r + \sqrt{r^2 - 4r(x - x_0)^2}}{2}}$$

- The value of δ may be positive or negative depending on the value of $Q'(x)$.

Letting $x = x_0$, we have the edge model located at x_0.

$$\begin{aligned}
\sigma &= +\sqrt{-Q'(x_0)/Q'''(x_0)}, \\
\delta &= \sqrt{2\pi}\sigma Q'(x_0), \\
v &= Q(x_0)/\,\mathsf{erf}\left(\frac{\alpha}{\sqrt{2}\sigma}\right) - \frac{\delta}{2},
\end{aligned}$$

The Edge-Based Super-Resolution Algorithm

Step 1 Choose one of the images as the reference image (the one with best lighting).

Step 2 Estimate the motions involved in the image sequence.

Step 3 Estimate the edges and blur models.

Step 4 Warp all the edges/blur models to the reference image and fuse them.

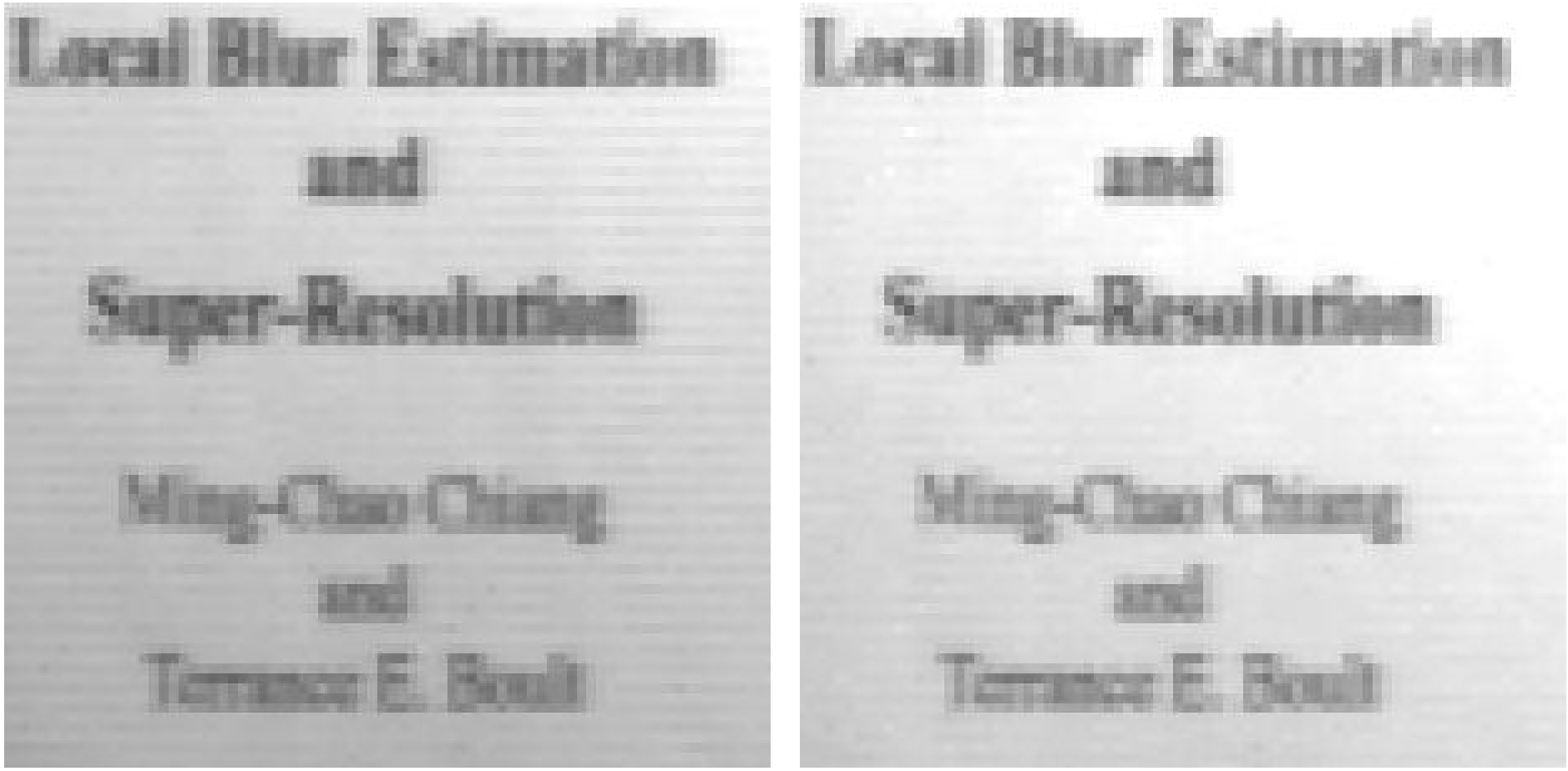
Step 5 Use the fused edge/blur models and the reference image to compute the super-resolution intensity image.

Step 6 Optional deblurring stage.

Reestimate the Underlying Image

Given the edge location x_0 and its corresponding edge model (v, δ, and σ), the underlying image can be reestimated as follows:

1. Replace the intensity values to the left of x_0 by v and the intensity values to the right of x_0 by $v + \delta$. The replacement is no more than three pixels to the left and right so that the algorithm remains local.
2. Compute $\bar{E}_i$ and $\bar{E}_{i+1}$ based on the intensity values given in the previous step.
3. Compute $\bar{Q}_i(x)$ based on $\bar{E}_i$ and $\bar{E}_{i+1}$ while at the same time the image is being scaled up.

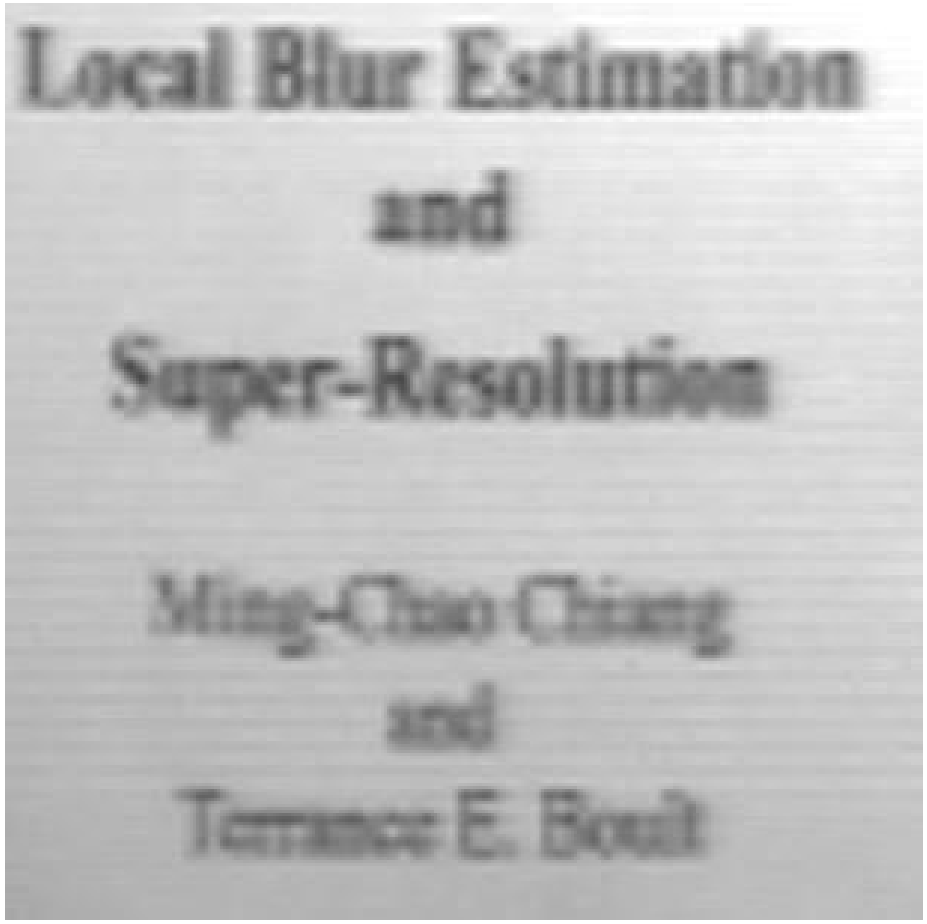


4X Pixel Replication

Two original images showing two different illuminations

4X Bilinear Resampling

Two original images showing two different illuminations



SR QRW w/o deblurring

SR QRW w/ deblurring

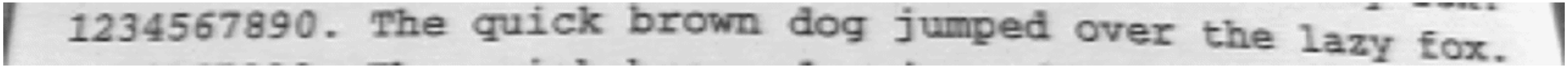
SR LBE w/o deblurring

SR LBE w/ deblurring

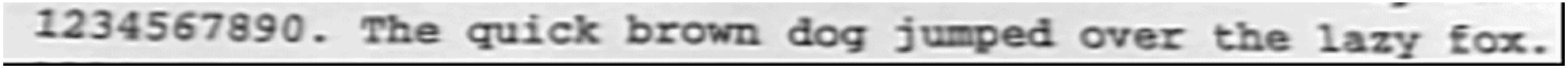
SR QRW w/ deblurring

SR LBE w/ deblurring

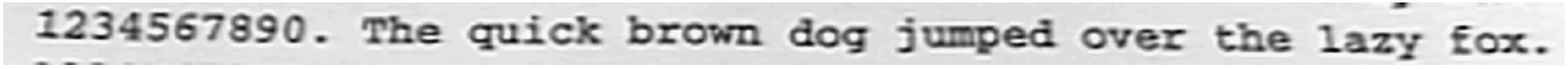
Quantitative Measurement Using OCR



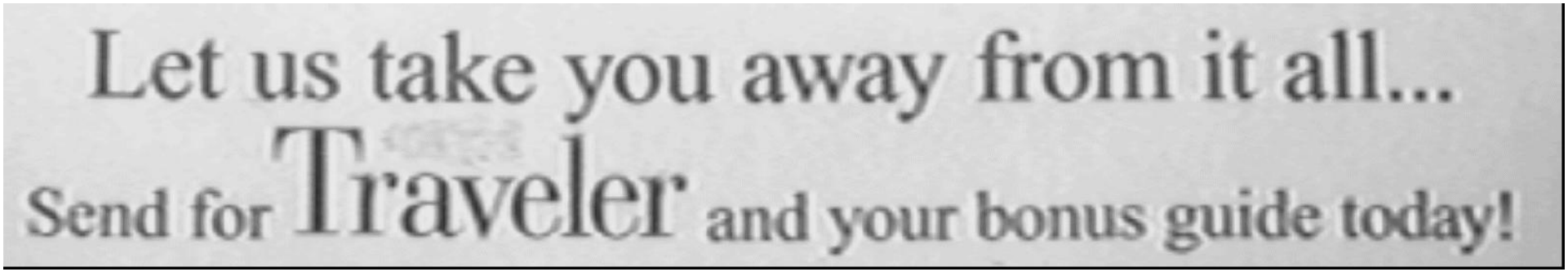
4X Bilinear Resampling w/o distortion correction



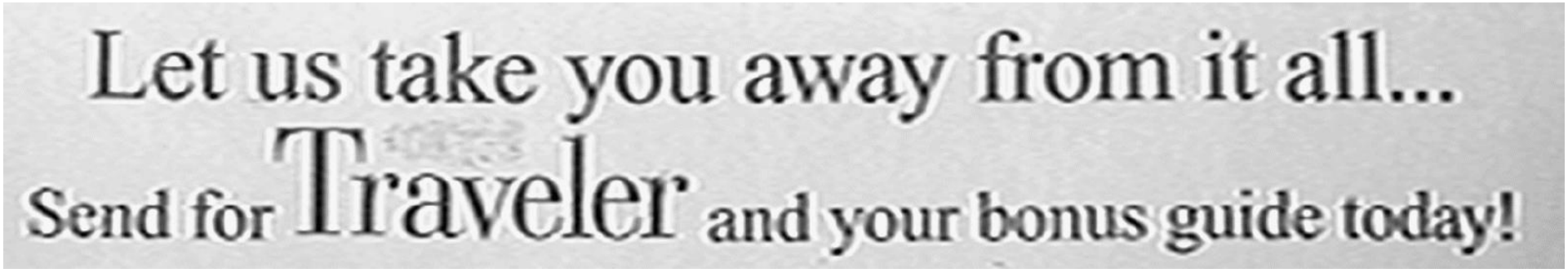
SR Bilinear w/ deblurring



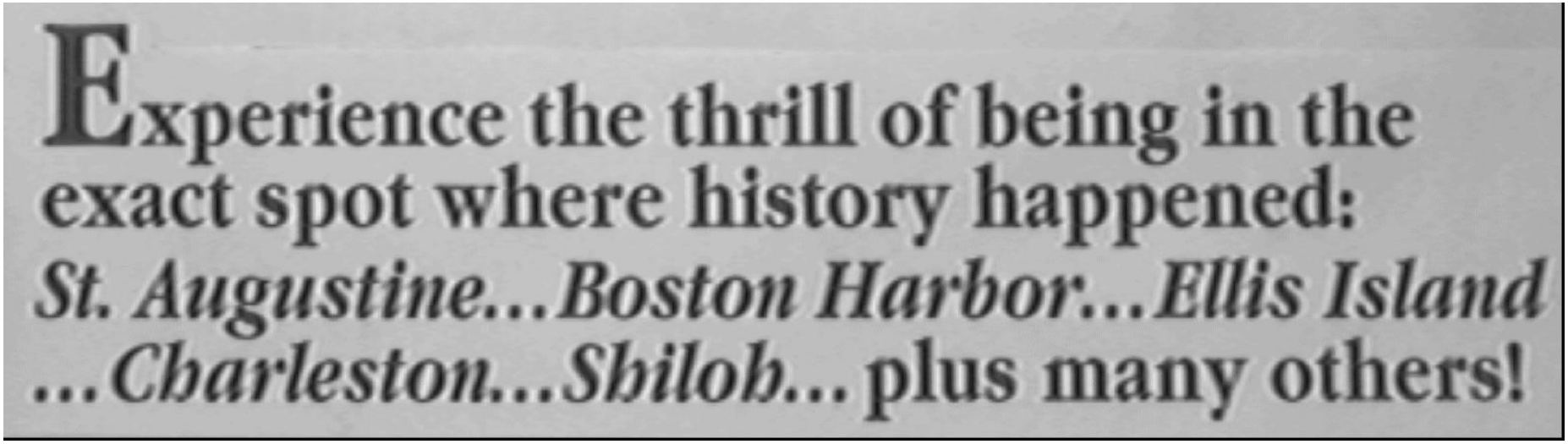
SR QRW w/ deblurring



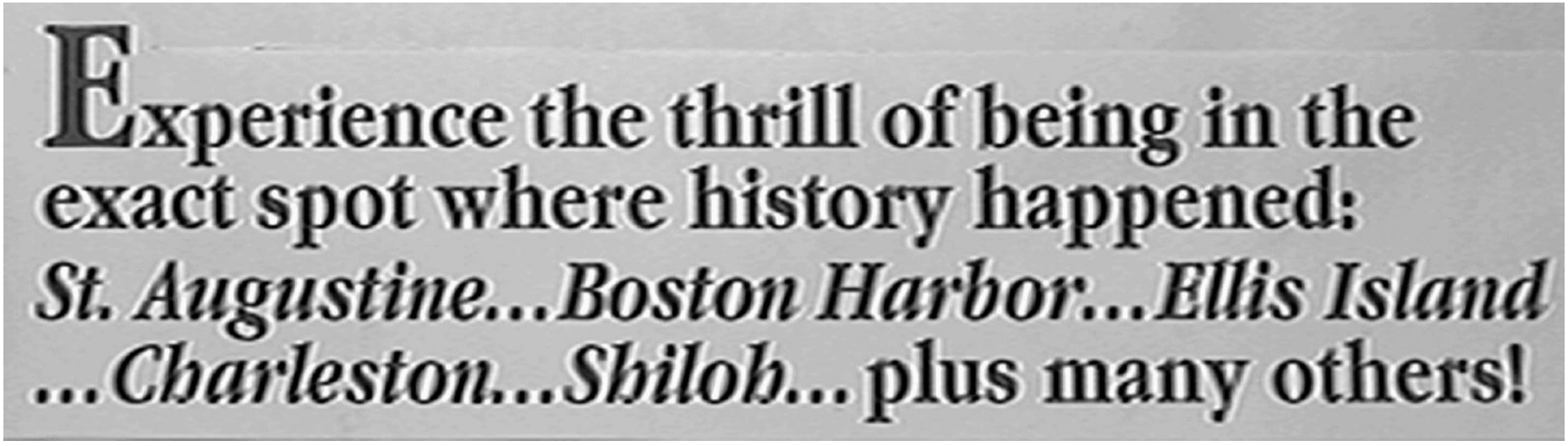
4X Bilinear Resampling (1720x300)



4X SR QRW w/ deblurring (1720x300)



4X Bilinear Resampling (1960x560)



4X SR QRW w/ deblurring (1960x560)

Quantitative Measurement Using OCR

Except for the first example,

- the images are normalized with respect to the super-resolution image with deblurring, and
- the same threshold is used to binarize all the images.

We use a character-oriented OCR program because

1. it is easy to make it quantitative, and
2. we consider this to be a good measure for applications such as license-plate reading and other pattern recognition problems.

Category	Characters	Visually	Occurrence
1	0, O	ambiguous	Fig. 1
2	1, l, !, |	ambiguous	Fig. 2; Fig. 3 lines 1 and 4
3	/, l, t	ambiguous	Fig. 3 lines 3 and 4
4	h, b	ambiguous	Fig. 3 line 4
-	Otherwise	distinguishable	

- C_i: the number of incorrectly recognized characters,
- C_m: the number of missing characters,
- C_e: the number of extra characters, and
- C_s: the number of characters that are split into two or more characters.

Quantitative Results

Method	Output of OCR
BRX	1234561990. te gu<clc bow dog jumped over te sazs
BRDCD	:23*;56?990. X:e quick brown dog jt:Wed over the Lazf for
SRDCD	2234567890. The quick brown dog jumped over the lazes 'ox.

Method	C_r^i	C_u^i	$\%C_r^i$	C_i	C_m	C_e	C_s	$C_{i+m+e+s}$
BRX	32	16	67	7	8	0	1	16
BRDCD	35	13	73	11	1	0	1	14
SRDCD	45	3	94	2	0	0	1	3

Fig. 1

Method	Output of OCR	C_i	C_m	C_e	C_s
BR	Let us take you assay from it all...	0	0	0	1
	Send fior TI aveler and your bonus gtude todyl	1	2	2	1
BRD	Let us take you assay from it all...	0	0	0	1
	Send for Tl aveler and your bonus guide today!	0	0	0	1
SR	Let us take you away from it all...	0	0	0	0
	Send fior Tl alreler and your bonus guide today	0	1	1	2
SRD	Let us take you away from it all...	0	0	0	0
	Send for Tra@&eler and your bonus guide today!	0	0	0	1

Method	C_r^i	C_u^i	$\%C_r^i$	C_i	C_m	C_e	C_s	$C_{i+m+e+s}$
BR	61	5	92	1	2	2	2	7
BRD	64	2	97	0	0	0	2	2
SR	63	3	95	0	1	1	2	4
SRD	65	1	99	0	0	0	1	1

Fig. 2

Method	Output of OCR	C_i	C_m	C_e	C_s
BR	Experlence the thrill of belag in the	3	0	0	0
	exact spot where history happened	0	1	0	0
	St. Augustfne...Boston Hahbor...gWs Istand	5	2	0	0
	...Cbarleston@..SbSlob... plus maay othersl	0	7	0	0
BRD	Experience the thrill of being in the	0	0	0	0
	exact spot where history happened:	0	0	0	0
	St. Augustfne. . .Boston Hahbor...E s Island	3	2	2	0
	... CbarZeston...Sbilob... plus Expeothersl	9	1	1	0
SR	Experience the thrill of belag in the	2	0	0	0
	exact spot where history happened	0	1	0	0
	Stg Augustine...Boston Hahbor...SWs Island	4	2	0	0
	...Cbarlesto@t...Sbf/ob.. plus many othersl	6	1	0	1
SRD	Experience the thrill of being in the	0	0	0	0
	exact spot where history happened:	0	0	0	0
	Sf. Augustxne...Boston Harbor...Sis Istand	4	2	0	0
	...Cbar/eston...Sbi/oh.. plus many othersl	5	1	0	0

Method	C_r^i	C_u^i	$\%C_r^i$	C_i	C_m	C_e	C_s	$C_{i+m+e+s}$
BR	124	18	87	8	10	0	0	18
BRD	128	14	90	12	3	3	0	18
SR	126	16	89	12	4	0	1	17
SRD	130	12	92	9	3	0	0	12

Fig. 3

Quantitative Measurement Using SLAM

The fundamental idea is to use

- appearance matching and pose estimation as the primary measure, and
- image-quality metric as a secondary measure.

We consider two image-quality metrics:

1. Per Pixel Error (PPE)
2. Normalized L_2 Norm (L_2^N)

Per Pixel Error (PPE)

It is defined to be the sum of the absolute difference between the scaled-up image and the original image divided by the number of pixels.

- The interpretation is complicated because the appearance matching algorithm does not use the original image, but a projection.
- It is unclear what exactly is the meaning of the PPE, or its impact on object recognition or pose estimation. The small images are well below the Nyquist sampling rate of the large images.
- This measure gives a reasonable qualitative indication of how similar a scaled-up image is to the corresponding high-resolution image in the learning set.

Normalized L_2 Norm (L_2^N)

It is defined to be the distance between the eigenspace projections of the scaled-up image and the original image divided by the number of eigenvectors.

- This is justified by the observation that if all eigenvectors are used, then the distance between the corresponding points in the eigenspace is exactly the L_2 norm.
- In our case, using all eigenvectors is too expensive; so we approximate it with a subspace projection.

Given the parametric eigenspace, the L_2^N norm can be computed as follows:

$$L_2^N = ||u - v||/m$$

where

- $u = [e_1, e_2, \ldots, e_m]^T [X_r - c]$ is the projection of the original image X_r,
- $v = [e_1, e_2, \ldots, e_m]^T [Y_r - c]$ is the projection of the scaled-up image Y_r,
- m is the number of eigenvectors,
- c is the average of all the images in the learning set, and
- r is the orientation or pose parameter in degrees.

Quantitative Measurement Using SLAM

Experimental Setup

0°

90°

180°

270°

Object image set 1

Image from Training Set

4X SR QRW w/ deblurring

(90°)

4X Cubic Convolution

4X SR QRW w/ deblurring

(90°)

Image from Training Set

4X SR QRW w/ deblurring

(180°)

4X Cubic Convolution

4X SR QRW w/ deblurring

(180°)

$0°$ $90°$ $180°$ $270°$

A case in which cubic convolution outperforms super-resolution.

Except at $0°$, most of the image content is undersampled or aliased high frequency text, and blurring these images improves the L_2^N estimation for both cubic-convolution and super-resolution. Super-resolution, which strives to improve the high-frequency content, does not help.

Quantitative Results

Object 1	16/128		32/128		PPE
	PE	L_2^N	PE	L_2^N	
X_{0°	0.0	0.0	0.0	0.0	0.000
CC_μ	5.6	292.1	5.6	163.8	0.132
CC_σ	0.0	6.7	0.0	5.0	0.000
SR_8	0.0	183.8	0.0	105.0	0.110
SR_{16}	0.0	184.0	0.0	105.2	0.110
SR_{32}	0.0	184.2	0.0	105.4	0.110
SRD_8	0.0	176.2	0.0	102.2	0.130
SRD_{16}	0.0	176.5	0.0	102.4	0.130
SRD_{32}	0.0	176.9	0.0	102.7	0.130
X_{90°	-0.5	0.0	-0.5	0.0	0.000
CC_μ	44.7	341.5	47.5	179.3	0.142
CC_σ	0.0	0.5	0.0	0.3	0.000
SR_8	33.5	227.7	0.0	121.6	0.120
SR_{16}	33.5	227.7	0.0	121.6	0.120
SR_{32}	33.5	227.6	0.0	121.5	0.120
SRD_8	33.5	232.0	0.0	124.7	0.140
SRD_{16}	33.5	231.9	0.0	124.7	0.140
SRD_{32}	33.5	231.8	0.0	124.6	0.140

Object 1	16/128		32/128		PPE
	PE	L_2^N	PE	L_2^N	
X_{180°	1.7	0.0	1.7	0.0	0.000
CC_μ	-46.2	427.3	-29.8	220.2	0.166
CC_σ	11.8	107.5	19.5	53.1	0.023
SR_8	-44.8	428.2	-5.6	217.7	0.160
SR_{16}	-44.8	429.5	-5.6	218.5	0.160
SR_{32}	-44.8	414.7	-5.6	211.1	0.150
SRD_8	-42.0	456.7	-5.6	231.7	0.170
SRD_{16}	-42.0	457.9	-5.6	232.5	0.170
SRD_{32}	-42.0	446.1	-5.6	226.6	0.170
X_{270°	1.1	0.0	1.1	0.0	0.000
CC_μ	-0.5	251.0	0.0	160.1	0.139
CC_σ	1.1	25.1	0.0	12.3	0.000
SR_8	0.0	169.2	0.0	113.4	0.130
SR_{16}	0.0	168.8	0.0	113.3	0.130
SR_{32}	0.0	168.3	0.0	113.2	0.130
SRD_8	0.0	154.6	0.0	102.4	0.140
SRD_{16}	0.0	166.8	0.0	108.1	0.140
SRD_{32}	0.0	166.2	0.0	107.7	0.140

Object 2	16/128		32/128		PPE
	PE	L_2^N	PE	L_2^N	
X_{0°	0.0	0.0	0.0	0.0	0.000
CC_μ	0.8	168.7	0.8	118.9	0.143
CC_σ	1.3	17.1	1.3	10.0	0.000
SR_8	0.0	134.4	0.0	97.5	0.130
SR_{16}	0.0	133.0	0.0	96.9	0.130
SR_{32}	0.0	132.8	0.0	96.8	0.130
SRD_8	0.0	131.7	0.0	95.4	0.150
SRD_{16}	0.0	130.4	0.0	94.9	0.150
SRD_{32}	0.0	130.1	0.0	94.9	0.150
X_{90°	-0.5	0.0	-0.5	0.0	0.000
CC_μ	2.0	243.9	2.0	172.2	0.149
CC_σ	1.3	51.3	1.3	25.3	0.013
SR_8	0.0	158.9	0.0	127.1	0.130
SR_{16}	0.0	166.2	0.0	129.9	0.130
SR_{32}	2.8	192.7	2.8	140.2	0.140
SRD_8	0.0	151.1	0.0	118.4	0.140
SRD_{16}	0.0	159.0	0.0	121.7	0.140
SRD_{32}	2.8	188.1	2.8	134.0	0.150

Object 2	16/128		32/128		PPE
	PE	L_2^N	PE	L_2^N	
X_{180°	-1.1	0.0	-1.1	0.0	0.000
CC_μ	2.4	183.4	3.1	155.7	0.141
CC_σ	1.0	22.1	1.8	15.8	0.000
SR_8	2.8	164.4	2.8	141.7	0.140
SR_{16}	2.8	165.0	2.8	142.1	0.140
SR_{32}	2.8	165.1	2.8	142.2	0.140
SRD_8	2.8	159.8	2.8	134.7	0.150
SRD_{16}	2.8	160.4	2.8	135.1	0.150
SRD_{32}	2.8	160.5	2.8	135.2	0.150
X_{270°	1.1	0.0	1.1	0.0	0.000
CC_μ	4.8	428.6	2.8	275.9	0.197
CC_σ	1.3	14.4	0.0	8.6	0.000
SR_8	2.8	381.2	2.8	254.3	0.190
SR_{16}	2.8	382.8	2.8	255.1	0.190
SR_{32}	2.8	383.0	2.8	255.3	0.190
SRD_8	2.8	391.7	2.8	261.4	0.210
SRD_{16}	2.8	393.4	2.8	262.4	0.210
SRD_{32}	2.8	393.5	2.8	262.5	0.210

Object 3	16/128		32/128		PPE
	PE	L_2^N	PE	L_2^N	
X_{0°	0.0	0.0	0.0	0.0	0.000
CC_μ	0.0	229.2	0.0	206.7	0.170
CC_σ	0.0	3.2	0.0	1.4	0.000
SR_8	0.0	191.5	0.0	189.8	0.170
SR_{16}	0.0	191.7	0.0	190.0	0.170
SR_{32}	0.0	191.7	0.0	190.1	0.170
SRD_8	0.0	190.1	0.0	189.2	0.190
SRD_{16}	0.0	190.4	0.0	189.5	0.190
SRD_{32}	0.0	190.5	0.0	189.6	0.190
X_{90°	-0.5	0.0	-0.5	0.0	0.000
CC_μ	2.8	346.4	2.8	221.6	0.172
CC_σ	0.0	17.6	0.0	8.7	0.000
SR_8	2.8	265.2	2.8	182.4	0.160
SR_{16}	2.8	266.5	2.8	182.9	0.160
SR_{32}	2.8	267.0	2.8	183.2	0.160
SRD_8	2.8	262.3	2.8	180.2	0.170
SRD_{16}	2.8	263.7	2.8	180.9	0.170
SRD_{32}	2.8	264.3	2.8	181.2	0.170

Object 3	16/128		32/128		PPE
	PE	L_2^N	PE	L_2^N	
X_{180°	-1.1	0.0	-1.1	0.0	0.000
CC_μ	0.0	162.2	-0.1	129.3	0.141
CC_σ	0.0	6.8	0.5	6.2	0.000
SR_8	0.0	140.9	0.0	114.4	0.140
SR_{16}	0.0	140.4	0.0	114.1	0.140
SR_{32}	0.0	140.4	0.0	114.1	0.140
SRD_8	0.0	144.8	0.0	115.8	0.150
SRD_{16}	0.0	144.1	0.0	115.4	0.150
SRD_{32}	0.0	144.1	0.0	115.4	0.150
X_{270°	1.1	0.0	1.1	0.0	0.000
CC_μ	2.2	357.1	2.9	227.4	0.181
CC_σ	1.2	50.4	0.5	27.3	0.012
SR_8	2.8	308.1	2.8	197.4	0.170
SR_{16}	2.8	305.8	2.8	196.2	0.170
SR_{32}	2.8	304.5	2.8	195.5	0.170
SRD_8	2.8	313.1	2.8	198.4	0.190
SRD_{16}	2.8	310.7	2.8	197.1	0.190
SRD_{32}	2.8	309.2	2.8	196.3	0.190

Object 4	$16/128_N$		$16/128_B$		$16/128_{B-N}$	
	PE	L_2^N	PE	L_2^N	PE	L_2^N
$X_{0°}$	0.0	0.0	0.0	0.0	0.0	0.0
CC_μ	-5.0	222.4	-5.0	235.8	0.0	6.0%
CC_σ	0.0	12.7	0.0	8.2	0.0	-4.5
SR_8	0.0	151.7	0.0	183.3	0.0	20.8%
SR_{16}	0.0	145.0	0.0	179.2	0.0	23.6%
SR_{32}	0.0	144.7	0.0	180.3	0.0	24.6%
SRD_8	0.0	163.0	0.0	179.1	0.0	9.9%
SRD_{16}	0.0	156.6	0.0	174.8	0.0	11.6%
SRD_{32}	0.0	155.8	0.0	175.7	0.0	12.7%
$X_{90°}$	-0.6	0.0	-0.6	0.0	0.0	0.0
CC_μ	1.4	210.3	3.1	199.6	1.7	-5.1%
CC_σ	1.4	1.0	2.6	1.2	1.2	0.2
SR_8	5.6	249.9	5.6	231.3	0.0	-7.5%
SR_{16}	5.6	256.7	5.6	238.9	0.0	-7.0%
SR_{32}	5.6	256.2	5.6	237.7	0.0	-7.2%
SRD_8	5.6	266.2	5.6	237.8	0.0	-10.7%
SRD_{16}	5.6	272.4	5.6	245.7	0.0	-9.8%
SRD_{32}	5.6	272.2	5.6	244.4	0.0	-10.2%

Object 4	$16/128_N$		$16/128_B$		$16/128_{B-N}$	
	PE	L_2^N	PE	L_2^N	PE	L_2^N
$X_{180°}$	-1.1	0.0	-1.1	0.0	0.0	0.0
CC_μ	-5.6	224.5	-5.6	204.6	0.0	-8.9%
CC_σ	0.0	3.3	0.0	3.0	0.0	-0.2
SR_8	-5.6	253.6	-5.6	223.9	0.0	-11.7%
SR_{16}	-5.6	255.1	-5.6	224.4	0.0	-12.0%
SR_{32}	-5.6	255.1	-5.6	223.3	0.0	-12.4%
SRD_8	-5.6	270.1	-5.6	233.3	0.0	-13.6%
SRD_{16}	-5.6	272.1	-5.6	233.8	0.0	-14.0%
SRD_{32}	-5.6	272.3	-5.6	232.8	0.0	-14.5%
$X_{270°}$	1.1	0.0	1.1	0.0	0.0	0.0
CC_μ	0.7	231.9	0.2	224.8	-0.5	-3.1%
CC_σ	1.2	2.7	0.7	1.9	-0.5	-0.8
SR_8	5.6	360.1	5.6	328.8	0.0	-8.7%
SR_{16}	5.6	362.9	5.6	330.7	0.0	-8.9%
SR_{32}	5.6	362.3	5.6	329.9	0.0	-8.9%
SRD_8	5.6	377.4	5.6	339.8	0.0	-10.0%
SRD_{16}	5.6	380.1	5.6	341.9	0.0	-10.1%
SRD_{32}	5.6	379.5	5.6	341.1	0.0	-10.1%

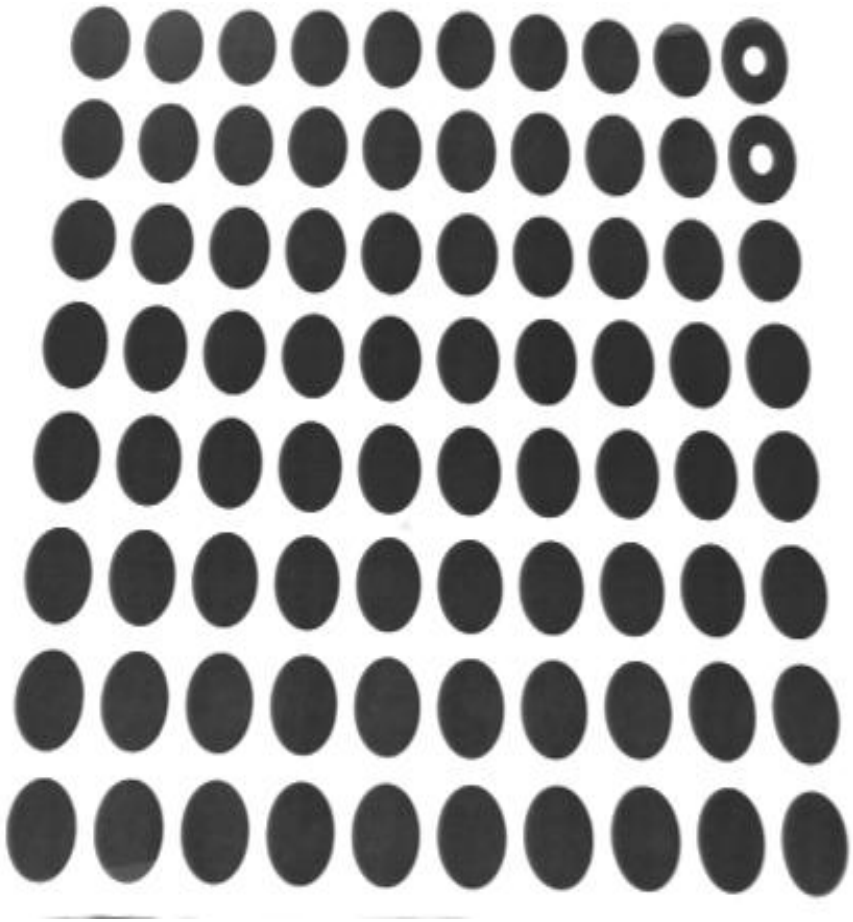
A case in which cubic convolution outperforms super-resolution.

Except at $0°$, blurring these images improves the L_2^N estimation for both cubic-convolution and super-resolution. Super-resolution, which strives to improve the high-frequency content, does not help.

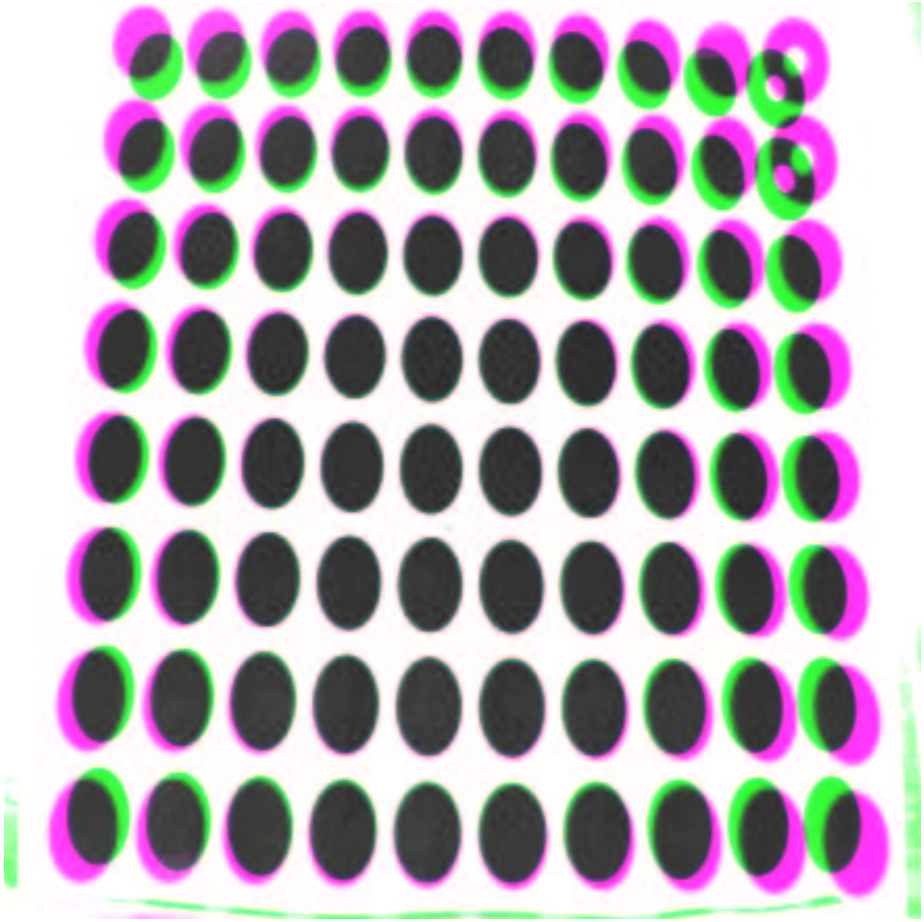
Quantitative Conclusion

- On average, using 32 eigenvectors, cubic convolution had a pose error of 8.12 degrees while super-resolution with 8 images, deblurring or not, had a pose error of 1.4 degrees.
- 32 eigenvectors give better results than 16.
- 8 images are generally enough.
- Deblurring may or may not help.

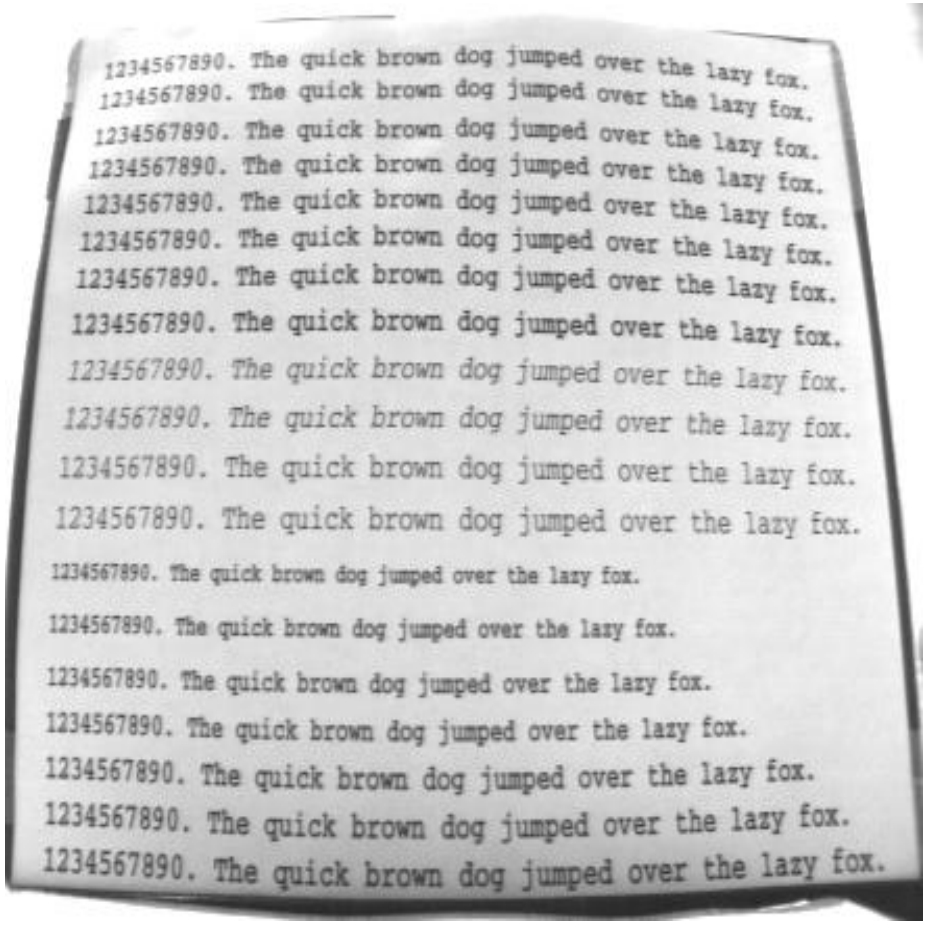
Camera Calibration and Distortion Correction



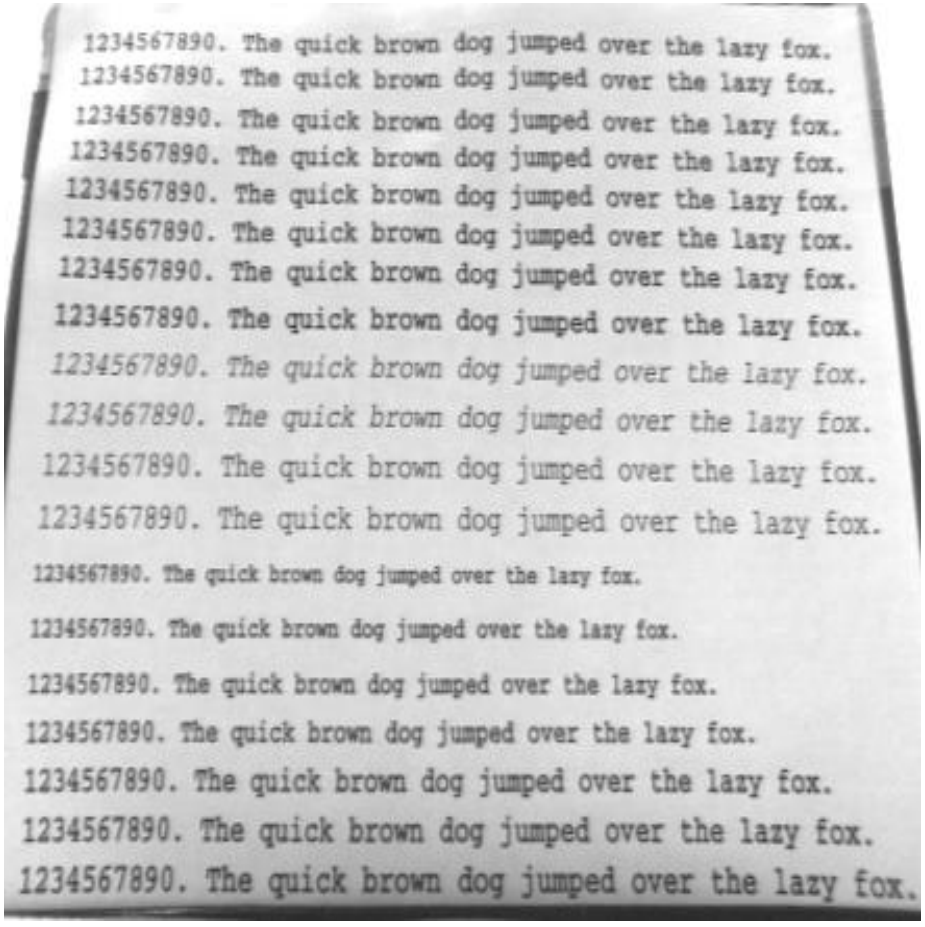
Corrected pattern



Corrected pattern on top of original



Original text image



Corrected text image

Left image

Right image

Corrected left image

Corrected right image

Conclusions

- Prove that if linear interpolation is used to determine E_i, the resulting imaging-consistent reconstruction algorithm QRR is tantamount to cubic convolution with the "optimal" value of $A = -.5$.
- The imaging-consistent warping algorithms (i.e., the integrating resamplers).
- Image-Based Super-Resolution
- Edge-Based Super-Resolution
- Quantitative Measurement Using OCR
- Quantitative Measurement Using SLAM