In 1994 Proc. of IEEE Asilomar Conf. on Signals, Systems, and Compu ters

AUTOMATED DESIGN OF TWO-DIMENSIONAL RATIONAL DECIMATION SYSTEMS

Brian L. Evans and *Jürgen Teich*

Dept. of Electrical Eng. and Comp. Sciences
University of California
Berkeley, CA 94720-1772 USA
E-mail: {ble,teich}@eecs.berkeley.edu

Christian Schwarz

Max-Planck-Institut fuer Informatik
Im Stadtwald
D-66123 Saarbruecken, Germany
E-mail: schwarz@mpi-sb.mpg.de

Abstract

The paper gives an automated procedure to design rational decimation compression systems that resample two-dimensional bandpass signals at their Nyquist rates. Our procedure takes a sketch of the desired bandpass shape in the frequency domain, circumscribes it with a parallelogram of minimal area, and linearly maps the minimal enclosing parallelogram onto one period of the frequency domain. From the rational matrix that performs the linear mapping, we compute the parameters of the upsampler, filter, and downsampler in the compression system. The compression system only has linear components.

1. Introduction

Decimation systems are used to reduce the amount of data for applications in which the crucial data occupies a certain frequency band. Images, for example, are often oversampled, so most of the signal energy resides at low frequencies. If the high-frequency content (edges, texture, etc.) is not important, then the image can be decimated to a lower spatial resolution. In video processing, two-dimensional decimators can be used to convert sequences of images from interlaced to non-interlaced format. These decimators preserve the frequency content in a diamond-shaped (i.e., parallelogram-shaped) passband centered at zero frequency. Seismic data is sampled in position and time [1]. Data falling on position-time lines having the same slope correspond to seismic waves having the same velocity since slope $= \frac{\Delta\text{position}}{\Delta\text{time}} =$ velocity. Fan filters are used to pass ranges of velocities [1, 2]. If the range of passed velocities is small, the fan filters produce "narrowband" signals. When the narrowband spectrum includes either frequency axis, the spectrum takes the shape of a periodically repeating parallelogram, as shown in Figure 1.

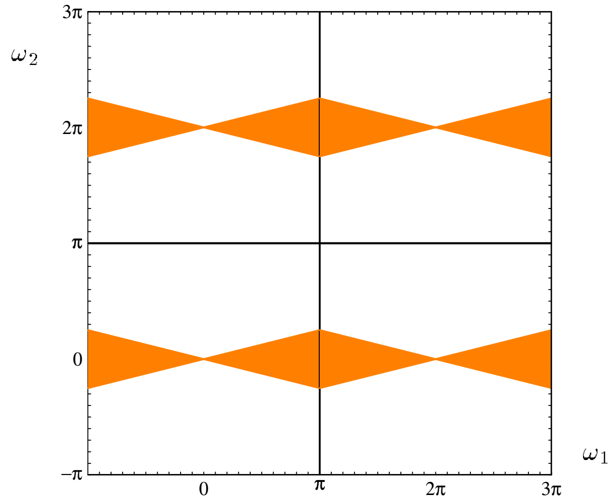


Figure 1: A Fan Filter Frequency Response

The paper discusses the automation of the design of compression systems that resample 2-D bandpass signals at their Nyquist rates. The compression sys-

J. Teich was supported by a grant from the German government (DFG). B. L. Evans was supported by, and J. Teich received supplemental support from, the Ptolemy project. The Ptolemy project is supported by the Advanced Research Projects Agency and the U.S. Air Force (under the RASSP program, contract F33615-93-C-1317), Semiconductor Research Corporation (project 94-DC-008), National Science Foundation (MIP-9201605), Office of Naval Research (via Naval Research Laboratories), the State of California MICRO program, and the following companies: Bell Northern Research, Dolby, Hitachi, Mentor Graphics, Mitsubishi, NEC, Pacific Bell, Philips, Rockwell, Sony, and Synopsys. World Wide Web information about the Ptolemy project is located at URL http://ptolemy.eecs.berkeley.edu. B. L. Evans can be reached by phone at +1 510 643-6686 and by fax at +1 510 642-2739.

C. Schwarz was supported by, and performed this research while he was a visiting scholar at, the International Computer Science Institute, 1947 Center Street, Suite 600, Berkeley, CA 94704-1198 USA. The International Computer Science Institute is an independent, non-profit research institute affiliated with the University of California campus in Berkeley.

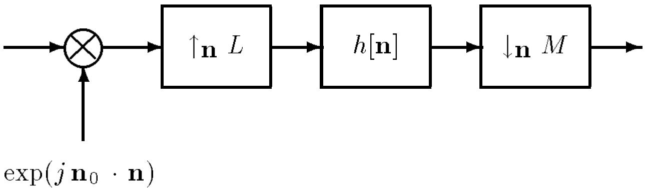


Figure 2: Flow Graph of a Two-Dimensional Decimator

tem consists of a cascade of a modulator, an upsampler, a filter, and a downsampler, as shown in Figure 1 above. The compression system performs a *rational resampling* of the input signal in the sense that the overall sampling rate is altered by a rational factor equal to $\frac{|\det M|}{|\det L|}$. That is, the sampling matrix of the input signal is mapped to the sampling matrix of the output signal by a matrix of rational numbers.

This paper presents an automated design procedure which designs the compression system based on a sketch of the input frequency band to be preserved. Section 2 reviews recent results in the theory underlying 2-D rational decimation systems. Section 3 shows how to automate the design of 2-D rational decimation systems. Besides synthesizing various theoretical results together, Section 3 introduces a new result in the field of computational geometry— finding a parallelogram with minimal area that circumscribes a convex polygon. Section 4 gives a design example using the our implementation in the Mathematica [3] symbolic mathematics environment. The Mathematica implementation is encoded in a set of signal processing packages by Evans, McClellan, and others [4, 5]. Section 5 concludes the paper and discusses areas of future research.

2. Background

Rational decimation systems extract a connected portion of the input frequency spectrum (the passband) and resample it at a lower rate. To resample the desired passband at the Nyquist rate, the passband is first shifted in the frequency domain so that it is centered at zero frequency and then mapped onto one period of the frequency domain while preventing aliasing and imaging effects. In this paper, we choose the period $\omega_1 \in [-\pi, \pi) \bigcup \omega_2 \in [-\pi, \pi)$, which is also known as the fundamental frequency tile. Recall that the frequency domain is periodic with period of 2π in each frequency variable.

If the passband is a parallelogram centered at zero frequency whose coordinates are rational multiples of π, then the corners of the parallelogram can be mapped one-to-one with the corners of the fundamental frequency tile [6]. The one-to-one mapping, which is carried out by a rational matrix, expands the interior of the parallelogram to fill the fundamental frequency tile. From the rational matrix, we can easily find the values for upsampling matrix L and the downsampling matrix M and specifications on the shape of the passband for the filter $h[\mathbf{n}]$ [6]. In fact, the rational matrix can always be factored to find L and M that are relatively prime [7]. If L and M are relatively prime, then efficient polyphase implementations always exist for these rational decimator designs [8]. The family of two-dimensional interpolated finite impulse response (IFIR) filters provides efficient realizations of the filter $h[\mathbf{n}]$ [9], and recently, a general design framework has been developed to design IFIR filters [10].

3. An Automated Design Procedure

The last section reviewed the theory to resample one family of bandpass signals at its Nyquist rate. The spectrum of the bandpass signals must have the shape of a parallelogram whose vertex coordinates are rational multiples of π. In this section, we derive a general procedure that can design rational decimation systems for the family of bandpass signals whose spectrum is a polygon, either concave or convex.

In developing an automated procedure to design rational decimators, we will find a procedure that takes a polygonal bandpass spectrum and finds the parallelogram with minimal areas that circumscribes the polygon and whose coordinates are rational multiples of π. The arbitrary passband shape will be represented as a polygon with N vertices. The 2π by 2π frequency domain is divided into a grid of $(2M+1)\times(2M+1)$ points. An exhaustive grid search procedure to find the parallelogram would require $\mathcal{O}(M^6 N^2)$ operations. Evans and Sakarya have presented a heuristic procedure that requires $\mathcal{O}\left(N^2\right)$ operations [7]. In this section, we derive a procedure that only requires $\mathcal{O}\left(N\right)$ operations and that is optimal if the vertices of the original polygonal bandpass spectrum are rational multiples of π. The latter condition is satisfied if a designer sketches a bandpass shape using a mouse because the polygonal vertices selected will be on a frequency grid.

We first find the convex hull of the input polygon, e.g. by using the Graham Scan method [11]. Next, we find the parallelogram of minimal area that contains convex hull (polygon). At the heart of finding the minimal enclosing parallelogram is our proof given in [12] that

(1) two adjacent sides of the minimal enclosing parallelogram overlap with two of the edges of convex polygon, and

(2) the other two adjacent sides of the minimal enclosing parallelogram intersect the convex polygon at two of the convex polygon vertices.

Conversely, of the two parallel sides of a candidate parallelogram, one side will intersect the convex polygon at an edge, and the other side at a vertex. A pair of two parallel sides of different slopes forms a parallelogram.

This observation suggests a simple algorithm that pairs together all possible edge-vertex pairs of the convex polygon, so there are $(N-2)+(N-3)+(N-4)+\ldots$ or $\mathcal{O}\left(N^2\right)$ combinations. This approach, though optimal, is quadratic in the number of polygon vertices N and unnecessarily considers many parallelograms. We can trim the parallelogram candidates based on an ordering relationship to produce an algorithm that is linear in N.

The formal description of the linear algorithm depends on the following definitions. Let e and v be an edge and a vertex of the convex polygon C. Let l_e be the supporting line of e, and let l_v be the line though v that is parallel to l_e. The supporting line of a line segment e is the line containing e, and the supporting line of a convex polygon C is a tangent of C. The pair (e, v) is called an *antipodal pair* of C if l_e and and l_v *support* C. Using this terminology, we have proven in [12] that there is a parallelogram P_C such that for both of its axes l_1 and l_2, the intersection with C consists of an antipodal pair. The algorithm also relies on a *projection* operator $P_l(f)$ which projects of f (an edge or vertex) onto the plane orthogonal to line l.

The linear algorithm is described to the right in Figure 3. A `C++` implementation is given in the Appendix of [12], and the next section gives an example of interaction with the Mathematica implementation.

Initialization: We compute a list $L = L_1, \ldots, L_N$ of all anti-podal pairs of C such that they are stored in clockwise order of their edges, each of which has a pointer to its anti-podal vertex. When there are parallel edges in C, then there are two adjacent anti-podal vertices and we will refer to the first one in the clockwise ordering.

Iteration: We move two edge pointers z_1 and z_2 clockwise around C. Each configuration of z_1 and z_2 represents a pair of anti-podal pairs, and the area of the corresponding enclosing parallelogram is computed. At each iteration, we move z_1 one vertex around the polygon and then move z_2 forward, so z_2 is always ahead of z_1 by 1 to $N-1$ vertices. We start by setting $z_1 = z_2 = 1$ and then repeat the following steps:

1. If $z_1 = z_2$, then advance z_2 by one. Let e be the edge in C pointed to by z_2, let v be its anti-podal vertex, and let l be the supporting line of the edge pointed to by z_1. Then, the projection $P_l(v)$ is a real number and $P_l(e)$ is an interval in $\mathcal{R}$. As long as $P_l(v) < P_l(e)$, i.e., the $P_l(v)$ is smaller than the lower boundary of $P_l(e)$, advance the pointer z_2.

2. If $P_l(v) \in P_l(e)$, the compute the corresponding parallelogram and keep track of the minimum computed so far. In the special case that that $P_l(v)$ intersects the lower boundary of $P_l(e)$, advance z_2 to the successor of e, say e', and do the same computation for the pair (e', v).

3. If $z_1 < N$, then advance z_1 by one and go to step 1; otherwise, stop.

Figure 3: Linear-Time Algorithm to Find the Minimal Enclosing Parallelogram of a Convex Polygon [12]

4. Design Example

Our Mathematica implementation of the automated design procedure is a combination of using the notebook interface to sketch the desired bandpass spectrum as a polygon and our `DesignDecimationSystem` routine that designs the decimation system from the polygon. The routine takes one argument that is either a polygon or a resampling matrix. Given a resampling matrix, the routine factors the resampling matrix into the up/downsampling matrices L and M. Given a polygon, the routine performs all of the steps in the procedure. The routine supports an option `Mod` which sets an upper limit on the denominator of the coordinates of the parallelogram computed in step 2 of the procedure. The rest of this section discusses the design of a two-dimensional decimator for a circular passband using our Mathematica implementation.

We ran the design procedure on a circular passband of radius 1. We approximated the circle with a twenty-sided polygon whose vertices were given by a simple Mathematica formula. The design procedure reported a packing efficiency of 79.2% and an 8-to-1 compression ratio. The best packing efficiency, 86.6%, is obtained by circumscribing the passband with a regular hexagon [1]. For non-linear compression, the theoretical upper limit on the compression ratio is 4π, which is approximately 12.5-to-1.

5. CONCLUSION

In some image, video, seismic applications, only a portion of the frequency content of the signals is important. This paper gives an automated procedure to design the 2-D compression system shown in Figure 1 to resample a desired bandpass signal at its Nyquist rate. This approach is based on the knowledge of where the important bandpass region resides in the frequency domain. The procedure circumscribes the desired bandpass spectrum with a parallelogram of minimal area and maximally decimates the parallelogram by computing

1. the convex hull of the desired bandpass spectrum,
2. the parallelogram whose coordinates are rational multiples of π, whose extent circumscribes the passband, and whose area is minimal,
3. the vector necessary to shift the center of the parallelogram to the origin (baseband) which is equal to the modulation parameter $\mathbf{n}_0$,
4. the rational matrix H that maps the parallelogram onto the fundamental frequency tile, and
5. the integer matrix factors L and M of the rational matrix H such that $H = L^{-1}M$.

We have implemented the procedure in the version 3.0 of the Signal Processing Packages and Notebooks [4, 5] for the computer algebra system Mathematica. Our implementation allows the designer to specify the desired bandpass spectrum either graphically by using a mouse or manually by calculating the vertices of the polygon. Our implementation, however, does not design the two-dimensional filter. An earlier version of our implementation is in version 2.9.5 of the Signal Processing Packages which is available by anonymous FTP to `gauss.eedsp.gatech.edu` (IP #130.207.226.24) in the directory `pub/Mathematica`.

There are three immediate areas of future research. One area is to include the design of the filter in the automated procedure, e.g., by adapting the procedures in [10]. A second area is to develop the theory for rational decimation systems based on a hexagonal bandpass spectrum since it would decimate circularly bandlimited signals more efficiently [1]. The issue of circumscribing a convex polygon bu a hexagon of minimal area is also an open research problem. A third area of future research is to extend this procedure to multiple channels to design perfectly reconstructing two-dimensional filter banks [13, 14] based on tiling the frequency domain into different convex polygon shapes.

6. References

[1] D. E. Dudgeon and R. M. Mersereau, *Multidimensional Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1984.

[2] R. Ansari, “Efficient IIR and FIR fan filters,” *IEEE Trans. on Circuits and Systems*, vol. CAS-34, pp. 941–945, Aug. 1987.

[3] S. Wolfram, *Mathematica: A System for Doing Mathematics by Computer*. Redwood City, CA: Addison-Wesley, second ed., 1991.

[4] B. L. Evans and J. H. McClellan, “Symbolic analysis of signals and systems,” in *Symbolic and Knowledge-Based Signal Processing* (A. Oppenheim and H. Nawab, eds.), pp. 88–141, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1992.

[5] B. L. Evans, L. J. Karam, K. A. West, and J. H. McClellan, “Learning signals and systems with Mathematica,” *IEEE Trans. on Education*, vol. 36, pp. 72–78, Feb. 1993.

[6] T. Chen and P. P. Vaidyanathan, “On the choice of rational decimation systems for multidimensional signals,” in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. V, (Minneapolis, MN), pp. 527–530, Apr. 1993.

[7] B. L. Evans and F. A. Sakarya, “Interactive graphical design of two-dimensional compression systems,” in *Second National Workshop on Signal Processing*, (Marmaris, Turkey), pp. 173–178, Apr. 1994.

[8] T. Chen and P. P. Vaidyanathan, “The role of integer matrices in multidimensional multirate systems,” *IEEE Trans. on Signal Processing*, vol. 41, pp. 1035–1047, Mar. 1993.

[9] T. Chen and P. P. Vaidyanathan, “Recent developments in multidimensional multirate systems,” *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 3, pp. 116–137, Apr. 1993.

[10] R. Manduchi, “Two-dimensional IFIR structures using generalized factorable filters,” Tech. Rep. UCB/CSD 93/785, Computer Science Division, Dept. of EECS, University of California, Berkeley, CA 94720, Oct. 1994.

[11] R. Sedgewick, *Algorithms*. Redwood City, CA: Addison-Wesley, 1988.

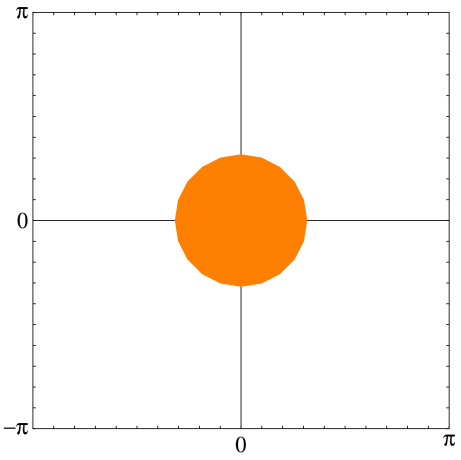
[12] C. Schwarz, J. Teich, E. Welz, and B. L. Evans, “On finding a minimal enclosing parallelogram,” Tech. Rep. TR-94-036, International Computer Science Institute, 1947 Center Street, Suite 600, Berkeley, CA, 94704-1198, Aug. 1994. Report available from the World Wide Web at URL `http://www.icsi.berkeley.edu` and from the gopher site `gopher.icsi.berkeley.edu` under `11/icsi/techreports`.

[13] R. H. Bamberger, *The Directional Filter Bank: A Multirate Filter Bank for the Directional Decomposition of Images*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, Dec. 1990.

[14] T. R. Gardos, *Analysis and Design of Multidimensional FIR Filter Banks*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, June 1993.

```
poly =
  Polygon[
    N[ Table[ { Cos[theta], Sin[theta] },
              { theta, Pi/10, 2 Pi, Pi/10 } ] ]
];

Show [ Graphics[ { RGBColor[1,1/2,0], poly } ],
       AspectRatio -> 1, Axes -> True,
       Frame -> True,
       FrameTicks -> { piTicks, piTicks },
       PlotRange -> {{-Pi, Pi}, {-Pi, Pi}} ]
```

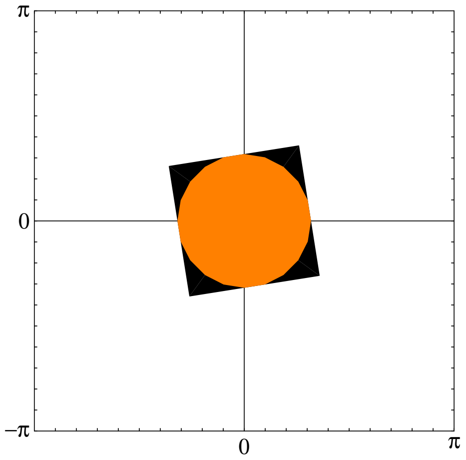



```
-Graphics-
```

This example finds the 2-D rational decimation system to resample a circular passband (with radius 1) near its Nyquist rate. The circular passband is approximated by a twenty-sided regular polygon whose vertices are generated by a formula. The packing efficiency of 79.2% is with respect to the polygon and not the circular passband itself. The rational decimator design achieves an 8-to-1 compression ratio ($|\det M|/|\det L| = 8$). The theoretical upper limit on the compression ratio is the ratio of the area of the fundamental frequency tile ($4\pi^2$) to the area of the twenty-sided polygon (approximately π, the area of the circle) which is approximately 12.5-to-1.

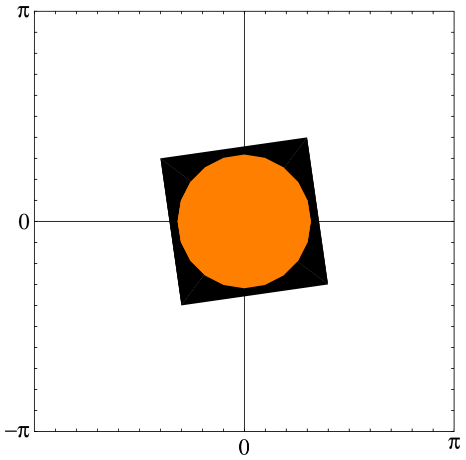
```
{ shiftVector, upMatrix, downMatrix } =
  DesignDecimationSystem[
    poly, Dialogue -> All, Mod -> 10 ]
Best packing efficiency with rotated rectangle

   having real-valued coordinates: 79.2%
```




```
Actual packing efficiency: 62.6%

  (out of a best possible 79.2%)
```




```
The compression ratio is 8-to-1.
{{0, 0}, {{5, 0}, {7, 1}}, {{2, 14}, {0, 20}}}
```

$$n_0 = (0, 0)$$

$$L = \begin{bmatrix} 5 & 0 \\ 7 & 1 \end{bmatrix}$$

$$M = \begin{bmatrix} 2 & 14 \\ 0 & 20 \end{bmatrix}$$

Figure 4: Automatic Design of a Decimator for Circularly Bandlimited Signals