

**VSIPL Forum Meeting Notes**  
**27 March 2001**  
**SPAWAR Systems Center, San Diego, CA**

Attendees:

James Lebak	MIT Lincoln Laboratory	jlebak@LL.mit.edu
Randall Judd	SSC-SD	judd@spawar.navy.mil
Josef Schaff	NAWCAD Pax River	SCHAFFJB@nawcad.navy.mil
Rick Pancoast	Lockheed Martin	rick.pancoast@lmco.com
Rob Ginn	NAWCAD Pax River	rob@sun701.nawcad.navy.mil
Win Smith	DNA Computing	wsmith@dna-cs.com
Dan Campbell	Georgia Tech	dan.campbell@gtri.gatech.edu
Anna Rounbehler	MSTI	arounbe@ieee.org
Dennis Cottel	SSC-SD	dennis@spawar.navy.mil
Howard Cohl	NAVO MSRC	HCOHL@navo.hpc.mil
James Kenny	Mercury	jfk@mc.com
Bill Harrod	SGI	harrod@sgi.com
Gary Boudreaux	MSTI	gary@mpi-softech.com
Keith Bromley	SSC-SD	bromley@spawar.navy.mil

Topics covered at this meeting:

- VSIPL 1.01 document
- Review of image processing draft
- VSIPL I/O functions
- Additional linear algebra functions
- User's group meeting

The forum's next meeting will be Tuesday, 21 August, 2001. Rick Pancoast of Lockheed-Martin volunteered to host the next meeting in Moorestown, NJ.

***VSIPL 1.01 document***

The forum reviewed the VSIPL 1.01 document sent out by Randy Judd. The following changes have been made:

- In the section "Introduction to VSIPL types", a paragraph has been added describing the "depth" and "shape" modifiers for function names. The forum had realized that such an explanation was lacking from the document.
- Function name "overloading" was removed from several of the "destroy" functions in the linear algebra chapter.
- The Kronecker product and cumulative sum functions were changed as noted in the minutes of the last meeting.
- Block find: Note has been added that it will fail when used on a derived block.
- Block rebind: No longer gives an error when used on a derived block. Note about derived blocks has been added.
- Block release: No longer gives an error when used on a derived block.

Gary Boudreaux pointed out that in the current spec, some function names include a “d” (depth) specifier, indicating that they could be complex, even though no complex versions of the function are defined in the spec. These functions include sq, sqrt, sumval, and sumsqval. Consensus of the forum was to keep the “d” in these names because complex functionality *could* be defined for all these functions.

After reviewing the spec the forum approved the document by a vote of 8-0. Voting institutions: MIT/LL, SSC/SD, NAWCAD, Lockheed-Martin, DNA Computing, Mercury, SGI, MSTI.

The forum noted that a few minor corrections to the profile documents need to be made as discussed on the mailing list. James Lebak has the action item to find out where the most recent versions of those are and make the necessary changes.

### ***Image Processing Review***

Win Smith had volunteered at the last meeting to review the image processing draft chapter. He stated that his goal was to focus on "mainstream" image processing functions. He reviewed the document with regard to functionality only and did not consider data structures.

Win's review was divided into several sections:

- Histogram operators
- Convolution-like operators, including
  - Convolution
  - Discrete differences
  - Resize operators (decimation and interpolation)
- Morphological operators
  - Ways of finding closed contours on an image — a good way to do automated shape finding
  - Win stated that these are not efficient on commercial processors because the operations are typically done a single bit at a time.
  - Rob Ginn stated that he knew of instances where these were multi-bit operators.
- Pad Operators
  - Ways to specify that an image should be extended at the edges
  - Could be incorporated into special instructions in the convolution and correlation create functions

Win recommended that the histogram, morphological, and pad operators be kept as they are. He also recommended that the current convolution and correlation operations from the VSIPL 1.01 document (which only work on floating-point types) should be expanded to handle integer data, and that a linear interpolation filter function could be added to handle the resizing operators. Following are some of the more important points of the ensuing discussion.

- The forum realizes that the next step is to define data structures. Unfortunately, no one has time to do this now.
- The sense of the forum is that a minimalist approach to image processing, based on integer matrices and tensors and integer versions of the convolution and correlation operators, is probably the right way to go. We are not going to be able to create an image processing library which is all things to all people; we want to be able to support those people who want to do image processing on their signal processing data.

- For now, such an approach probably need only support monochrome, possibly with the addition of RGB supported through integer 3-tensor views. This would be a change from the scheme proposed in the current draft, which uses a new image block type.
- There may be issues with re-using the current image processing document, because it has a copyright in the name of Khoral Research Inc. (KRI). Bill Harrod took the action item to call the folks at KRI to find out what their status is and whether they feel any claim to the current spec.
- Keith Bromley pointed out that Brian Sroka may be able to provide input on what functions the Common Imagery Processor (CIP) project requires. This would give us a sense of how well the proposed functionality would fit the needs of one particular program.

Anna Rounbehler asked whether anyone in the forum had looked at how VSIPL relates to the Texas Instruments “DSP Express” environment. No one had, but all agreed that it would be valuable to do so if someone had the time to do it.

### ***I/O Functions***

Randy Judd reviewed his latest document for I/O functions — functions to allow us to copy, save, and communicate opaque VSIPL objects. It was quickly noticed that the functions in his latest document don’t match those in the minutes of the last meeting, and Randy agreed to change the document to reflect the interface defined in those minutes (as modified by anything in these minutes).

There are four I/O functions: import, export, find, and size. The import function generated the most discussion. The basic characteristics of the import function are as follows.

- Import takes as an argument an old QRD object and a memory area. The intent is to enable the implementation to re-use some resources associated with the old QRD object.
- To create a new object, NULL is passed in place of the old QRD object.
- Import essentially “admits” the memory area passed into it. Data in that area cannot be modified by the user again until a destroy has been done on the object, or until the same object is passed into the import function again with a different memory buffer.

An implementation of a QRD might consist of two memory areas; a descriptor area and a data area. Implementations are free to choose whether both areas are part of the imported buffer or if only the data area is part of the imported buffer.

- If these areas are both part of the imported buffer, the import function will not have to do any memory allocation.
- If the imported buffer consists only of the data area, the first call to import will have to allocate memory to store the descriptor area.
- In subsequent calls to import, the new QRD object can be passed in and memory for the descriptor area need not be allocated. This was deemed to be acceptable.
- In the case of an application reusing a single buffer in a loop and importing a QR each time through that loop, the application will have to call import and destroy during each iteration. If the descriptor area is not part of the imported buffer, this will lead to memory allocation in the working loop. The forum agreed this was a problem but saw no easy resolution. [Editor’s note: perhaps the “notes” section of the man page should mention this problem and recommend double-buffering.]

For the export and size functions, consensus of the forum was that the man pages should specify that the sizes are given in bytes.

For a specific example (the QRD object), there are two ways to create objects now; the old way using the create function, and the new way using the 'import' function.

- Import only works on objects created by the import function (not on those created by the vsip\_qrd\_create function).
- The export function must work on objects created in both ways.
- Find should return null when used on objects created by vsip\_qrd\_create
- Size should work on both types of objects.

### ***Additional VSIPL Functionality***

Bill Harrod reviewed two new functions proposed for being able to extract singular values corresponding to particular singular values. Either right or left singular values (the "U" or "V" matrices) can be extracted, and the vectors extracted can correspond to any range of singular values. The forum seemed to be pleased with these functions.

A short discussion ensued about the "scope" of the next version of the VSIPL standard. Consensus of the forum at this time is that it consists of a small number of focussed extensions rather than major new areas. In the view of the chair, this is reflected in the designation of the next version as "VSIPL 1.1" rather than "VSIPL 2.0". Randy Judd would like to have this next version, whatever it is called, finished by the time his funding runs out on 30 September.

One function that several people thought would be desirable was a 'windowed FFT' function. We could add a create for a new object, that included an additional parameter for the window vector. Multiple FFT ("mfft") versions of this function with particular windowing functions across rows or columns would also be desirable.

### ***User's Group Meeting***

The forum is still interested in hosting a "User's group" meeting. However, owing to Mark Richards' schedule, the time has been changed from early summer to the fall. The forum expanded the proposed outline for the user group meeting, and assigned responsible parties for different pieces of it.

The general concept is that this would be a one and a half to two day meeting, with the first day having a more tutorial nature and the second day allowing users to share their experiences. Vendors were asked to contact customers and ask whether any of them would have particular experiences that they would like to share at a forum like this. The intent is to have a few talks lined up to be able to advertise them at the HPEC workshop in September.

Day 1:

Introduction	(0.5 hr)	Chairs
Design Principles:	(0.5 hr)	Chairs
Object based		
Relationship to FPS, vendor libraries		
Speed vs Portability		
Profiles		

Tutorials

General overview	(1 hr)	Randy Judd or James Lebak
Blocks, views, memory, Admit/release, data spaces		

(Tutorials Continued)

VSIPL Interfacing example	(0.5 hr)	Jamie Kenny
Signal Processing and Linear Algebra	(1.5 hr)	Bill Harrod and Win Smith
Example application walkthrough	(1 hr)	Randy Judd or James Lebak

Day 2:

User Experiences

- Invited presentations from known customers
- Lockheed-Martin performance results
- Others (someone suggested Brian Sroka of MITRE might have some results)

Product overviews

Vendor posters

Future development	(0.5 hr)	Chairs
--------------------	----------	--------

Those responsible for particular talks should send a draft of their talks to the VSIPL core list by the beginning of June, and prepare draft viewgraphs to bring to the VSIPL meeting in August.