

О программе Tiff-16 Data Analysis

Кон В. Г., 19-08-2025, <http://kohnvict.narod.ru>

Программа выполняет серию операций обработки экспериментальных данных, записанных в картинке tiff формата, которые создает 2D координатный детектор. Программа написана на языке *vkac1* и сделана в стандарте программ общего назначения (с номерами). В самом начале она открывает файл для редактирования входных данных. Все входные данные одного варианта записываются в одну строку, но таких строк может быть много. Выбор варианта реализуется при установке курсора в первую или вторую позицию выбранной строки перед выходом из редактора по нажатию клавиши [esc]. Работа над данной программой началась в процессе работы над научной статьей номер 266 по списку научных публикаций на моем сайте.

Все операции формально независимы и могут выполняться в любом порядке. Номер операции является первым параметром входных данных. От его значения зависит структура всех остальных параметров. Для разных номеров число других параметров и их смысл разные, хотя иногда есть повторения. По факту данная программа является коллекцией разных программ, собранных вместе и позволяющих проводить анализ экспериментальных данных, получаемых методом фазово-контрастной микроскопии на источниках синхротронного излучения.

Иногда используется сокращение МЛЯП, которое означает (Мой Личный Язык Программирования). Сама рассматриваемая программа является входной информацией для программы-интерпретатора МЛЯП. И именно эту программу, то есть ПИ-МЛЯП исполняет Операционная Система. Папка ПИ-МЛЯП называется главной папкой, то есть ГППИ-МЛЯП или просто ГППИ.

В некоторых операциях используется термин (карта цветов). Это последовательность цветов, которая соответствует интервалу значений аргумента. Каждое число в этом интервале имеет свой цвет, соответствие можно узнать из специальной полосы цветов, на которой проставлена шкала чисел. На данный момент программа имеет 8 карт цветов, номера начинаются от 0 и кончаются 7. Выбор карты цветов можно выполнять эмпирически, а всекарты цветов можно посмотреть на сайте <http://kohnvict.ucoz.ru/pic/colmaps.png>.

Число возможных операций в программе зависит от ее версии. Описание всегда соответствует самой последней версии. Но некоторые старые версии могут не выполнять кое-что из того, что записано в описании. Версия программы и описания определяется датой. Дата описания стоит под заголовком. Дату программы можно узнать по дате создания файла с адресом (s/XW/code/tiff16.ac1) относительно ГППИ.

♦ **1-я** операция определяет параметры tiff файла, в котором могут быть записаны полезные числовые данные. Формат tiff аналогичен формату pdf и был разработан для описания документов любой природы даже раньше, чем pdf. Он имеет очень сложную структуру записи, но которая легко определяется по цепочке (как бы раскручивается). То есть каждая новая информация сообщает где искать следующую. Так как вся информация записывается в нечитабельном виде, то необходима программа, которая все это конвертирует в читабельный вид.

Главная цель данной операции -- показать тот факт, что данный файл содержит черно-белую картинку, каждый пиксель которой записывается 1-м (в диапазоне от 0 до 255) или 2-м (в диапазоне от 0 до 65535) байтами без сжатия. Картинки в формате png имеют малый интервал от 0 до 255 для трех цветов в каждом пикселе, а формат jpg вообще искажает информацию и не годится для записи научных данных.

По этой причине операция показывает только те параметры (таги, tag), которые полезны для получения ответа на указанный выше вопрос. Главные таги сообщают ширину и высоту

картинки, способ записи и место размещения матрицы чисел. Остальные теги интереса не представляют, хотя тоже могут содержать какую-то информацию.

Входные данные очень простые – нужно указать папку относительно ГППИ или полный путь, начинающийся с буквы диска. Путь не должен содержать пробелов и должен быть окружен пробелами. Программа показывает менеджер файлов, начиная с этой папки для выбора нужного файла. После этого она показывает параметры этого файла. Эту информацию она записывает в файл (tif-inf.txt) в ту же папку, где находился tiff файл, и как раз его содержимое показывает на экране в редакторе текстов. **Пример:** 1 task/o7t/ E

♦ **2-я** операция вынимает из тифф файлов матрицу чисел, записанных в формате 16 бит или 8 бит, и записывает их в стандартный формат float языка программирования Java. В этом случае нужно указать адрес исходной папки, адрес папки для результатов и число (0 или 1). Программа просматривает все файлы в первой папке, находит те, которые удовлетворяют условиям, и вынимает из них числовую матрицу. Ее она записывает в файл с тем же именем, но с расширением .dat во вторую папку. В частном случае имена папок могут совпадать, тогда все происходит в одной папке.

Если число равно 0, то все происходит молча за исключением показа анимации уровня проделанной работы. Если файлов много, то после каждого обработанного файла длина линии проделанной работы увеличивается. Такая анимация на английском называется (progress bar). А если число равно 1, то указываются имена файлов, которые программа забраковала. Это иногда бывает полезно. Имена папок надо писать или относительно ГППИ или начиная с буквы диска.

Важное замечание. После того, как матрица вынимается из файла, она инвертируется по второму индексу. Это связано с тем, что мои программы рисуют картинки на осях координат и первые строки соответствуют вертикальной оси и находятся внизу. А стандартные программы по какой-то непонятной причине рисуют картинки сверху вниз. И детектор записывает матрицу как раз для такого показа. Приходится это исправлять.

Пример: 2 task/o7t/ task/o7t/ 1 E

♦ **3-я** операция работает с числовыми матрицами в формате Java, полученными из tiff файлов. Она использует две числовые матрицы в какой-либо папке. Используются только папки внутри ГППИ с учетом того, что на длину имени папки есть ограничение 42 символа. Сначала прочитывается первый файл и по требованию записывается его картинка. Затем прочитывается второй файл и по тому же требованию записывается его картинка.

Потом первая матрица делится на вторую, записывается результат в файл и картинка записывается по второму требованию. Наконец, новая матрица усредняется расчетом ее свертки с матрицей, отличной от нуля в небольшом квадрате в середине. И результат снова записывается в файл, а затем картинка записывается без условий. Это и есть главный результат операции, хотя бывает полезна и первая новая матрица.

Входные данные описываются формулой: ИРП И1Ф И2Ф И3Ф И4Ф n1 n2 n3 n4 , где ИРП – имя рабочей папки относительно ГППИ, И1Ф – имя 1-го файла без расширения. Файл с таким именем и расширением .dat должен существовать в указанной папке, а файл с таким именем и расширением .png будет создан, И2Ф – то же самое для второго файла, И3Ф – то же самое для файла после деления. Создаются оба файла с расширением .dat и .png. Условие на расчет картинки другое. И4Ф – то же самое после усреднения (свертки). Картинки программа не показывает, их надо посмотреть после работы программы другими средствами, в том числе и программами, написанными на МЛЯП.

Числовые параметры означают следующее: n1 – число точек матрицы по горизонтали, n2 – число точек матрицы по вертикали, n3 – число точек квадрата при усреднении. Наконец, n4 – это число с 4-мя разрядами типа 1001. Левый разряд, если 1, указывает на обращение контраста, второй слева указывает на номер карты цветов, второй справа указывает на

создание и запись картинок первых двух файлов, а правый, если 1 – на создание и запись картинки третьего файла. В момент записи текста можно было использовать 8 карт цветов с номерами от 0 до 7. Нули слева писать не обязательно.

Если указанный вторым файл не существует, то деление не производится и второй файл, естественно, не показывается. Если $n_3 = 0$, то усреднение не проводится. Если не выполняется и первое и второе, то программа просто показывает картинку первого файла и все.

Пример: 3 task/o7t/ 103 bg19 103bga 103bgb 2304 2304 9 611 E

♦ **4-я** операция работает с одной числовой матрицей. После считывания матрицы из файла она ограничивает область значений этой матрицы и новую матрицу снова записывает в файл. То есть задаются два числа А и В. Если значение элемента матрицы меньше А, то оно заменяется на А, а если оно больше В, то оно заменяется на В. Новая матрица рисуется на картинке.

Входные данные описываются формулой: ИРП И1Ф И2Ф n_1 n_2 р А В, где ИРП – имя рабочей папки с разделителем в конце, И1Ф – имя исходного файла без расширения, где записана матрица, И2Ф – имя нового файла, который будет создан без расширения. 5 чисел после имен файлов, которые необходимо записать – это размерности матрицы n_1 и n_2 , параметр показа картинки р и числа А и В, про которые написано выше. Параметр р -- это число с 2-мя разрядами типа 01. Левый разряд, если 1, указывает на обращение контраста, правый указывает на номер карты цветов.

Вообще говоря в этой операции размерности матрицы не используются. Можно было бы указать размерность вектора $n_1 * n_2$. Но иногда это число оказывается очень большим и удобнее все-таки указать два числа.

Пример: 4 task/o7t/ 103bga 1030 2304 2304 6 0.96 0.975 E

♦ **5-я** операция работает с одной числовой матрицей и делает то же самое, что и 3-я операция в самом конце. Она усредняет матрицу, но другим способом. Усреднение идет по строкам, как и раньше. В данном случае каждая точка строки, начиная со 2-й, заменяется на сумму из трех точек с весом, а именно $F_{\text{new}}(i) = F(i)/2 + [F(i-1) + F(i+1)]/4$. А первая и последняя точки в конце заменяются на вторую и предпоследнюю. После прохождения всех строк матрица транспонируется и процедура повторяется еще раз, а затем матрица снова транспонируется в исходное состояние. Интересно, что такую операцию можно проделывать много раз последовательно усредняя матрицу все более и более.

Входные данные описываются формулой: ИРП И1Ф И2Ф n_1 n_2 р n_p , где ИРП – имя рабочей папки с разделителем в конце, И1Ф – имя файла исходной матрицы И2Ф – базовое имя новых файлов, которые будут созданы. 4 числа после имен файлов, которые необходимо записать – это размерности матрицы n_1 и n_2 , параметр показа картинки р и число файлов n_p . Параметр р -- это число с 3-мя разрядами типа 101. Левый разряд, если 1, указывает на запись картинки из матрицы исходного файла после прочтения файла, второй слева указывает на обращение контраста, если равен 1, а правый указывает на номер карты цветов.

А параметр n_p содержит два числа в одном. Два разряда слева указывают на номер, который приписывается к базовому имени при записи матриц и картинок, А два разряда справа указывают сколько раз процедура будет исполнена. Номер приписывается двумя разрядами и 0 слева записывается, то есть вместо 1 будет 01. А вот во входных данных нули слева можно не писать. **Пример:** 5 task/o7t/ 1030 103 2304 2304 106 105 E

♦ **6-я** операция работает с одной квадратной числовой матрицей. После считывания матрицы из файла находятся координаты точки с максимальным значением, выбирается из матрицы ее часть в виде квадрата с меньшим размером, логарифмируется по условию и рисуется картинка с указанием на осях реальных координат части в исходной матрице.

Входные данные описываются формулой: ИРП ИФ n m р, где ИРП – имя рабочей папки с разделителем в конце, ИФ – имя исходного файла без расширения. Файл с таким именем и

расширением .dat должен существовать в указанной папке, а файлы с таким же именем, к которому добавляется буква m в конце, и расширениями .dat и .png будут созданы. Это и есть результат работы данной операции. Числа n и m – это размеры исходной квадратной матрицы и ее квадратного фрагмента. А число p имеет 5 разрядов. Левый указывает добавление к размеру шрифта 20 на картинке без масштабирования, если необходимо. На маленьких картинках после масштабирования шрифт может быть мелким и его можно увеличить до 29. Второй слева указывает число точек картинки на один пиксель. Средний, если 1, указывает на логарифмирование. Второй справа указывает на обращение контраста, если равен 1. Правый указывает на номер карты цветов.

Пример: 6 task/o7t/ 102bgb 2304 200 13010 E

♦ **7-я операция** делает то же самое, что и 6-я, только координаты точки с максимальным значением не ищутся, а фрагмент рисуется с центром в точке, координаты которой задаются во входных данных. Соответственно входные данные имеют еще два числа n1 и n2 в конце, которые и определяют эти координаты.

Пример: 7 task/o7t/ 102bgb 2304 200 4010 999 999 E

♦ **8-я операция** работает с несколькими файлами в заданной папке. В этих файлах находятся числовые квадратные матрицы, полученные из tiff файлов 16 бит. Для каждого входного файла она записывает в выходной файл одну строку текста. В этой строке указывается имя файла, минимальное, максимальное и среднее значения в матрице этого файла, затем после многоточия -- координаты первых наиболее высоких максимальных значений в этой матрице и само это значение.

Во входных данных после номера операции надо задать: Имя рабочей папки относительно ГППИ. Имя выходного файла без расширения, оно автоматически будет (txt). Общее имя серии файлов (не более 10) в этой папке. Файлы должны иметь числа последним символом в имени, в котором будет номер, начиная от 0 и не более 9. Расширение таких файлов быть (dat), но его писать не надо. Общее имя указывается без этих чисел. А также 4 числа: число файлов от 1 до 10, начальный номер файла (отсчет от нуля), число максимальных значений и размер матрицы по любому из измерений так как операция работает только с квадратными матрицами.

Число максимальных значений может быть любым, но все они записываются в одну строку. Если файлов 2 и более, то графически показывается процесс обработки файлов в процессе работы, то есть сколько сделано и сколько осталось. Координаты точек указываются в направлении сверху вниз. Так как мой МЛЯП рисует координаты снизу вверх, то перед расчетом реально делается инверсия матрицы по второму индексу. Такая операция позволяет выявить в короткой серии файлов наличие плохих пикселей, которые могут испортить контраст картинки для такой матрицы. У таких файлов максимальные значения случайные и разбросаны по всей матрице.

Пример: 8 task/o7t/ out 10 2 2 6 2304 E 1 s/XW/1/ E

♦ **9-я операция** работает с одной числовой матрицей. Она определяет минимальное, среднее и максимальное значения в каждой строке матрицы и три кривые рисует на графике. Для ее работы надо задать имя исходного файла без расширения, число точек по горизонтали и вертикали и число из двух разрядов, каждый из которых может иметь значение 0 или 1.

Левый разряд определяет тип график. Если 0, то график стандартный с фиксированной шириной, а если 1, то ширина графика зависит от числа точек в строке. Матрица рисуется так, что каждая ее строка имеет свой пиксель. При этом число точек может быть больше размера экрана. В этом случае появляется линейка прокрутки и все можно посмотреть.

Правый разряд определяет транспонирование матрицы. Если 0, то исходная матрица, а если 1, то транспонированная. В этом случае выдается информация о столбцах исходной

матрицы. Имена файлов для графиков получаются из имени исходного файла добавлением букв *m0* для исходной матрицы и *m1* для транспонированной.

Такая операция полезна по той причине, что карты цветов не дают полного представления о значениях в матрице. Хорошо показываются только координаты. А данная операция дает представление о значениях, но статистически.

Пример: 9 task/o7t/103 2304 2304 11 E

♦ **10-я** операция работает с одной квадратной числовой матрицей, которая была записана в tiff файл, 16 бит. Она позволяет выделить из этой числовой матрицы значения вдоль какой-либо линии, то есть одномерный массив из двумерного массива. Линия задается парой координат ее начала и конца. Если матрица имеет размерность *nx* и *ny*, то координаты точек начала и конца линии должны находиться в интервалах от 0 до *nx*–1 для *x*-координаты и от 0 до *ny*–1 для *y*-координаты. При этом координаты могут быть не обязательно целыми. Расчет проводится двумя методами. Первый использует интерполяцию и делает более точный и гладкий результат но считает немного дольше. Второй для каждой точки на линии берет ближайшую точку в матрице. Так расчет делается быстрее, но годится только для больших матриц с мелким шагом и гладких функций.

Входные данные содержат имена файлов относительно ГППИ с исходной матрицей без расширения (расширение всегда *.dat*), и с результатом без расширения (расширения *dat* и *png*) и 8 чисел, а именно *nx ny x1 y1 x2 y2 n m*, где *nx ny* – размеры исходной матрицы, *x1 y1 x2 y2* – координаты начальной и конечной точек линии, *n* – число точек на линии, *m* – номер метода расчета (1 или 2).

Пример: 10 task/o7t/103 s/103s1 2304 2304 1495.3 107.3 1575.3 187.3 81 1 E

♦ **11-я** операция работает с одной квадратной числовой матрицей, которая была записана в tiff файл, 16 бит. Она позволяет выровнять фон и убрать шум, оставляя только изображения объектов на ровном фоне. Идея такого преобразования основана на свойстве фазово-контрастного изображения не менять интегральную интенсивность. Метод хорошо работает в тех случаях, когда контраст от объектов существенно больше шума, а размеры объектов малы по сравнению с размерами на которых падающий пучок меняется заметным образом.

Процедура состоит в том, что вся матрица разбивается на части и в каждой части вычисляется среднее значение, после чего все значения этой части делятся на это значение и среднее значение становится равным 1. После такой обработки все части возвращаются на свои места, но теперь среднее значение всей картинке равно 1. И около этого среднего значения интенсивность меняется с большой амплитудой вблизи объектов и с малой амплитудой из-за шума.

Следующая операция делается со всей матрицей. А именно все значения, которые отличаются от 1 на величину меньше, чем заданное число, существенно меньше 1, заменяются на единицу. А к остальным значениям добавляется константа так, чтобы среднее значение оставалось равным 1. Картинка исходной матрицы рисуется по требованию, а картинка результат всегда.

Входные данные такие: ИФ *n m E p*, где ИФ – имя файла с матрицей относительно ГППИ, *n* – линейный размер (число точек) матрицы, *m* – размер квадратной части матрицы, *E* – величина отклонения значений от 1 для удаления шума (существенно меньше 1), *p* – трехразрядное число KRP, в котором *K* – ключ для создания картинки исходной матрицы, если равно 1, *R* – ключ для обращения карты цветов, если равно 1, *P* – номер карты цветов, может принимать значения от 0 до 7. **Важное условие.** Параметр *m* обязательно должен быть делителем числа *n*, иначе правильный результат не получится. Так в случае *n* = 2304 параметр *m* может принимать значения 2, 4, 8, 9, 16, 32, 64, 128, 256.

Пример: 11 task/o7t/1/110 2304 32 0.05 106 E

♦ **12-я** операция делает то же самое, что и 11-я операция, но организована иначе. А именно, она работает сразу со всеми файлами в указанной папке. При этом она выбирает для работы, только те файлы, размер которых соответствует входным данным. Остальные файлы она пропускает. В правильных файлах она выравнивает фон и убирает шум, а результат, то есть `dat` и `png` файлы с таким же именем, что и исходный файл, она записывает в другую папку. Папка реально должна быть другой, если необходимо сохранить исходные файлы, иначе они будут перезаписаны и пропадут.

Что касается картинок, то в стандартном режиме работы они создаются без карты цветов и без осей координат. При этом, в той же папке, где они создаются, записывается файл с названием (`mima.txt`), в котором указываются минимальное, максимальное и среднее значение для каждого из обработанных файлов. Среднее значение всегда 1, а вот минимальное и максимальное различаются. Но если параметр `p` (см. далее) имеет знак минус, то карта цветов и оси координат рисуются. В остальном используется модуль параметра `p`.

По этой причине при составлении большой картинке с мозаикой из этих файлов контраст на разных фрагментах будет разным, хотя среднее значение и будет совпадать. Правильный контраст, одинаковый для всех фрагментов делает другая операция, и это контраст зависит от того, какие конкретно файлы входят в общую картину мозаики. При этом максимальный контраст будет только для одного фрагмента из многих.

Входные данные такие: ИПИ ИПР `n m E p`, где ИПИ – имя папки источников относительно ГППИ, ИПР – имя папки результатов, `n` – линейный размер (чисто точек) матрицы, `m` – размер квадратной части матрицы, `E` – величина отклонения значений от 1 для удаления шума (существенно меньше 1), `p` – 4-хразрядное число MKRP, в котором `M` – ключ для создания картинок матриц с единичным фоном, если равно 1, `K` – ключ для создания картинок исходных матриц, если равно 1, `R` – ключ для обращения карты цветов, если равно 1, `P` – номер карты цветов, может принимать значения от 0 до 7.

Важное условие. Параметр `m` обязательно должен быть делителем числа `n`, иначе правильный результат не получится. Так в случае `n = 2304` параметр `m` может принимать значения 2, 4, 8, 9, 16, 32, 64, 128, 256. **Пример:** 12 pro/1/2/ pro/1/3/ 2304 32 0.05 1106 E

♦ **13-я** операция является дополнением в 12-й операции. Она использует матрицы, которые создает эта операция и тоже работает в пакетном режиме. Главная ее работа – создание рисунков этих матриц в одном и том же контрасте. То есть контраст всех картинок соответствует минимальному и максимальному значениям, которые задаются входными данными. Эти значения можно выбрать из таблицы файла с названием (`mima.txt`), который создает 12-я операция.

Такая операция полезна при составлении большой картинке с мозаикой, в которой картинки из разных файлов являются отдельными частями с правильным расположением внутри большой картинке. При этом получается правильное соответствие контрастов от разных дефектов в разных фрагментах.

В стандартном режиме работы картинки создаются без карты цветов и без осей координат. Но если параметр `p` (см. далее) имеет знак минус, то карта цветов и оси координат рисуются. В остальном используется модуль параметра `p`.

Входные данные такие: ИПИ ИПР `n m1 m2 p`, где ИПИ – имя папки источников относительно ГППИ, ИПР – имя папки результатов, `n` – линейный размер (чисто точек) матрицы, `m1` – минимальное значение для контраста, `m2` – максимальное значение для контраста, `p` – 2-хразрядное число RP, в котором `R` – ключ для обращения карты цветов, если равно 1, `P` – номер карты цветов, может принимать значения от 0 до 7.

Пример: 13 pro/1/3/ pro/1/4/ 2304 0.6 1.4 06 E

♦ **14-я** операция может использовать результаты 13-й операции, но может работать и полностью самостоятельно. Она создает картинку, которая аккумулирует информацию из большой серии картинок, размещенных в одной папке. На этой картинке все файлы из папки,

кроме двух, представлены в уменьшенном варианте и сопровождаются текстом в виде первых двух символов в названии файла. Для правильной работы программы в папке не должно быть лишних файлов. При этом порядок файлов на вновь создаваемой картинке определяется их местом в каталоге файлов. Наиболее существенными являются первые два символа.

По этой причине разумно в первые два символа записывать номера файлов, как они должны быть установлены согласно координатам, записанным в файл. Остальные символы в имени файлов могут быть любыми. Программа способна обработать не более 99 файлов. Если нужно больше, то программу придется перестроить. Но такого числа вполне хватит для многих вариантов. Важно также знать, что после файлов, которые будут представлены иконками, нужно определить еще один файл, который содержит картинку, на которой будут располагаться все иконки. А также нужно определить текстовый файл, в котором должны быть записаны координаты смещения всех иконок от левого нижнего угла полной картинки. Записывать надо целые числа в текстовом произвольном формате. Название этого файла может быть любым, но он в каталоге должен быть последним. То есть после предыдущего файла.

Входные данные такие – ИРП W H w h n S s r g b, где ИРП – имя рабочей папки относительно ГППИ, W и H – размеры полной картинки (фоновая картинка масштабируется на этот размер), w и h – размеры иконок (картинки в файлах масштабируются на этот размер), n – число файлов, которые будут обработаны (это число может быть меньше полного числа файлов в папке, но не может быть больше, так как оно принудительно уменьшается), S – размер текстов на иконках (если оно равно нулю, то текстов не будет), s – смещение текстов из левого нижнего угла иконки, r g b – компоненты цвета для текстов (красный, зеленый и синий в интервале от 0 до 255). Картинка результата записывается в ГППИ с названием mar.jpg. Потом ее можно переименовать и перенести куда следует.

Пример: 14 pro/1/1/ 1200 800 375 250 8 20 20 0 255 255 E

♦ **15-я** операция работает после выполнения 14-й операции, которая готовит иконки для серии файлов с картинками на специальной картинке, которая записывается в ГППИ с именем mar.jpg. Сами картинки находятся в папке, на которую указывают входные данные операции. Данная операция использует ту же самую папку с картинками, и тот же самый файл, который однако может быть переименован и перенесен в другую папку, но отличную от папки с картинками.

Цель данной операции – показывать исходные картинки при клике на их иконки на сборной картинке с исходным именем mar.jpg. То есть данная операция ничего не создает, она просто позволяет пользователю увидеть более подробно те оригиналы, иконки которых он видит с самого начала. Для этого надо просто кликнуть на иконке мышкой. И на том же месте экрана появится полная картинка. Если она больше размеров экрана, то окно будет иметь линейки прокрутки.

Клик на большой картинке вернет программу в исходное состояние. Можно выбрать другую иконку и так далее. Для окончания работы данной операции необходимо кликнуть какое-либо место вне иконок, а если такого нет, то на крестик в правом верхнем углу окна. Очевидно, что данная операция использует ту же самую папку с файлами, что и 14-я операция и плюс еще файл, который 14-я операция создает. Дополнительно нужно указать размеры иконок и их число. Соответственно входные данные имеют вид – ИРП ИФ w h n, где ИРП – имя рабочей папки относительно ГППИ, ИФ – имя файла с иконками относительно ГППИ, w и h – размеры иконок, n – число иконок (это число может быть меньше полного числа файлов в папке, но не может быть больше, так как оно принудительно уменьшается). И оно должно соответствовать числу иконок на картинке.

Большую картинку для просмотра делает сама программа специальной операцией. И при этом она показывает размер картинки, координаты курсора и цвет пиксела под курсором. Однако в системе Виндовс есть возможность показывать большую картинку также

средствами самой системы. Для этого надо задать параметр *n* со знаком минус. На работу программы знак не влияет. Он просто является указателем, что большие картинки будет показывать программа, которая в операционной системе привязана к данному расширению файлов. Если привязать *jrg* и *png* расширения к браузеру Яндекса, то он и будет их показывать. При этом он первоначально масштабирует картинку на окно браузера, а при клике мышкой – переходит в режим показа в полном размере.

Пример: 15 pro/1/1/ task/map/001.jpg 375 250 8 E

♦ **16-я** операция работает с одной квадратной числовой матрицей, которая была записана в *tiff* файл, 16 бит. Она определяет параметры области с центром в точке с максимальным значением. Первоначально предполагалось, что матрица показывает картинку с фокусировкой, в которой есть максимум с конечной шириной.

Входные данные должны указать имя файла без расширения относительно ГППИ, расширение всегда *.tif*, а также три числа: порядок байтов (0 для Low Byte Second, Motorola, и 1 для Low Byte First, Intel), число байтов заголовка и размер квадратной матрицы. В результате операция просто показывает 5 чисел: *x* и *y* координаты максимума, значение максимума и размеры пика по *x* и *y* осям (ширине и высоте). Значения параметров можно узнать прочитав таги по операции 1. **Пример:** 15 task/o7t/103 1 256 2304 E

♦ **99-я** операция является уникальной и имеет совсем другие статус и применение. Она сама ничего не делает. Ее цель другая. Дело в том, что основная программа сделана таким образом, что операции надо выполнять последовательно, одну за другой. Это удобно для начинающих, так как при такой работе обрабатываются кое-какие ошибки входных данных и любую операцию можно повторять много раз.

Но для рутинной работы это не удобно, так как заставляет все время быть у компьютера. Данная операция позволяет запускать любую комбинацию других операций в автоматическом режиме. То есть можно приготовить много строк в файле входных данных и запустить эту операцию. И она выполнит все варианты, указанные в этих строках последовательно и по любому сценарию.

Ее входные данные должны указывать число строк, которые надо выполнить и номера этих строк. Строку, в которой находится данная операция указывать нельзя. По этой причине разумно ставить эту операцию в самую нижнюю строку и указывать только те строки которые выше. И еще. Любая ошибка приведет к остановке всего процесса выполнения цепочки операций. Поэтому проверять входные данные надо особенно тщательно. Разумно все операции сначала пройти вручную и только потом использовать данный режим.

Пример: 99 2 2 12