

УДК 519.688: 520.88

Дата подачи статьи: 19.07.15

DOI: 10.15827/0236-235X.111.080-087

## **ОПТИМИЗАЦИЯ ПРОЦЕССА ОБНАРУЖЕНИЯ ОРБИТ НОВЫХ КОСМИЧЕСКИХ ОБЪЕКТОВ С ПОМОЩЬЮ ПАРАЛЛЕЛЬНОГО РАСЧЕТА ВОЗМОЖНЫХ ОРБИТ**

*(Работа выполнена в рамках НИР «Методы и программные средства разработки параллельных приложений и обеспечения функционирования вычислительных комплексов и сетей нового поколения», № 01201354596)*

*Е.А. Трушкова, д.ф.-м.н. ведущий научный сотрудник, katerinatr@mail.ru  
(Институт проблем управления им. В.А. Трапезникова РАН,  
ул. Профсоюзная, 65, г. Москва, 117997, Россия);*

*Г.А. Матвеев, ведущий инженер-исследователь, gera@prime.botik.ru  
(Институт программных систем им. А.К. Айламазяна РАН,  
ул. Петра I, 4а, г. Переславль-Залесский, 152021, Россия)*

Наблюдение и каталогизация малоразмерных объектов космического мусора в околоземном космическом пространстве требуют совершенствования применяемых методов и алгоритмов получения и обработки информации для повышения эффективности работы оптических средств. Один из подходов оперативного обнаружения и определения орбит некаталогизированных объектов космического мусора апробирован и успешно применяется на российских оптических наблюдательных наземных средствах. Однако много времени тратится на операцию полного перебора коротких серий измерений (треков) и выбор разнесенных во времени трех треков, предположительно, относящихся к одному объекту космического мусора и используемых для последующего построения возможной первоначальной орбиты. Данная статья посвящена вопросу сокращения перебора треков посредством предварительного анализа полученных измерений с помощью априорного построения границ изменения параметров возможных орбит. Вычислительные эксперименты, проводимые в рамках исследования этой прикладной задачи, направлены на уменьшение средней скорости обработки одного трека при увеличении точности полученной оценки. Скорость обработки одного трека естественным образом ограничена, так как, являясь частью общей программы обработки траекторных измерений, построение оценки параметров возможных орбит для каждого трека не должно сильно влиять на общее время цикла обработки измерений. При этом уменьшение средней скорости обработки одного трека и одновременное повышение точности оценок становятся возможными при параллельной организации работы программы. Так, для оптимизации работы программы обнаружения и определения орбит новых космических объектов предлагается параллельный алгоритм построения границ изменения параметров возможных орбит для последующего выбора троек треков, допустимые области изменений параметров которых имеют непустое пересечение. Алгоритм реализован на языке T++ для T-системы с открытой архитектурой (OpenTS).

**Ключевые слова:** космический мусор, околоземное космическое пространство, первоначальная орбита, параллельный алгоритм, динамическое распараллеливание, язык программирования T++, OpenTS, T-система, параллельное расширение Си++.

Основным направлением развития и применения оптико-электронных средств наблюдения космического пространства является расширение их возможностей по наблюдению, сопровождению и каталогизации малоразмерных объектов космического мусора (ОКМ) [1–5]. Примерами успешно развиваемых ключевых разработок в этом направлении являются сокращение временного цикла обзора одной площадки, применение эффективных стратегий обзора, оперативное обнаружение и определение орбит некаталогизированных ОКМ во время проведения сеанса наблюдений [6–8]. Одновременно с основным режимом работы обзорного средства работает и комплекс программ обработки траекторных измерений (ОТИ), осуществляющий оперативную обработку измерений для идентификации с орбитами каталогизированных космических объектов (КО). При успешной идентификации измерений комплекс программ ОТИ производит уточнение орбит каталогизированных КО по полученным измерениям. Если полученные измерения не идентифицировались ни с одним каталогизированным объектом, непосред-

ственно в процессе наблюдений дополнительно решаются задачи обнаружения и сбора информации по некаталогизированному ОКМ.

При получении очередной серии неидентифицированных измерений запускается задача определения орбит новых КО, которая заключается в поиске неидентифицированных измерений, относящихся к одному КО, построению и записи в каталог орбиты по этому КО.

Один из подходов оперативного обнаружения и определения орбит некаталогизированных ОКМ апробирован и успешно применяется на российских оптических широкопольных средствах [1]. Однако много времени тратится на операцию полного перебора коротких серий измерений (треков) и выбор разнесенных во времени трех серий, предположительно, относящихся к одному КО, используемых для последующего построения возможной орбиты.

Данная статья посвящена вопросу сокращения перебора треков при помощи предварительного анализа полученных измерений с точки зрения параллельного априорного построения границ из-

менения параметров возможных орбит. Подобный подход был использован в работе [9] для решения задачи идентификации орбиты с помощью генерации множества возможных орбит, построенных в параллельном режиме по имеющимся угловым измерениям от оптических наземных средств. Так, для ускорения работы программы обнаружения и определения орбит некаталогизированных ОКМ предлагается использовать вспомогательный алгоритм: строить для каждого трека допустимые области изменений трех параметров возможных орбит в виде пространственного параллелепипеда и затем выбирать только те тройки треков, соответствующие параллелепипеды для которых имеют непустое пересечение.

**Постановка задачи**

Для каждого трека (группы последовательных измерений)  $T_i$  данной последовательности треков  $T = \{T_i\}_{i=1, N}$  определяются границы изменения параметров возможных эллиптических орбит: дальности  $\rho_i \in [\rho_{i \min}, \rho_{i \max}]$ , наклонения  $I_i \in [I_{i \min}, I_{i \max}]$  и долготы восходящего узла  $\Omega_i \in [\Omega_{i \min}, \Omega_{i \max}]$  в заданной области изменения эксцентриситета  $e_i \in [e_{i \min}, e_{i \max}]$ . Далее находят все тройки треков  $(T_{i_1}, T_{i_2}, T_{i_3}) \in T$ , такие, что

$$\left( [\rho_{i_1 \min}, \rho_{i_1 \max}] \times [I_{i_1 \min}, I_{i_1 \max}] \times [\Omega_{i_1 \min}, \Omega_{i_1 \max}] \right) \cap \left( [\rho_{i_2 \min}, \rho_{i_2 \max}] \times [I_{i_2 \min}, I_{i_2 \max}] \times [\Omega_{i_2 \min}, \Omega_{i_2 \max}] \right) \cap \left( [\rho_{i_3 \min}, \rho_{i_3 \max}] \times [I_{i_3 \min}, I_{i_3 \max}] \times [\Omega_{i_3 \min}, \Omega_{i_3 \max}] \right) \neq \emptyset,$$

то есть границы изменения параметров возможных орбит для трех треков пересекаются (пересечение соответствующих параллелепипедов в пространстве изменения трех переменных  $(\rho, I, \Omega)$  не пусто).

Это позволяет в последовательности треков, переданных на обнаружение, существенно сократить количество перебираемых троек.

Так, в нижеследующем примере для входящей последовательности из 13 треков ( $n=13$ ) число полного перебора троек  $C_{13}^3 = 286$ , а проведенный предварительный анализ треков позволил сократить число перебираемых троек до 13.

Основная задача состоит в нахождении по измерениям (время  $t$ , прямое восхождение  $\alpha$ , склонение  $\delta$ , координаты измерительной станции  $R$ )

одного трека  $T_i = \{t, \alpha, \dot{\alpha}, \delta, \dot{\delta}, R, \dot{R}\}_{j=1, n}$  границ параметров орбиты, которой может принадлежать трек, при заданных интервалах изменения большой полуоси  $a_j \in [a_{j \min}, a_{j \max}]$  и эксцентриситета  $e_j \in [e_{j \min}, e_{j \max}]$ : границ возможного изменения дальности  $\rho \in [\rho_{\min}, \rho_{\max}]$ , наклонения  $I \in [I_{\min}, I_{\max}]$ , долготы восходящего узла  $\Omega \in [\Omega_{\min}, \Omega_{\max}]$ . Подобная задача рассматривалась в работе [9] и может

быть разбита на последовательное решение следующих выделенных вспомогательных задач.

**Вспомогательные задачи**

**Задача 1.** Выбор из измерений трека двух «хороших».

Для решения задачи значения  $\alpha_j$  и  $\delta_j$  приближаются кривыми второго порядка  $\alpha(t)$  и  $\delta(t)$  по методу наименьших квадратов. Из всех измерений данного трека выбираются два  $j_1, j_2$ , для которых расстояние между точкой трека  $(\alpha_j, \delta_j)$  и ее аппроксимацией  $(\alpha(t_j), \delta(t_j))$  наименьшее, при условии, что  $|j_1 - j_2|$  не меньше половины общего количества измерений. Далее, формально полагая  $j_1=1, j_2=2$ , выбранные измерения будем обозначать через  $(t_1, \alpha_1, \dot{\alpha}_1, \delta_1, \dot{\delta}_1, R_1, \dot{R}_1), (t_2, \alpha_2, \dot{\alpha}_2, \delta_2, \dot{\delta}_2, R_2, \dot{R}_2), t_1 < t_2$ .

**Задача 2.** По данным  $(\alpha, \delta, R), [a_{\min}, a_{\max}], [e_{\min}, e_{\max}]$  осуществляется построение начальных границ дальности  $\rho \in [\rho_{0 \min}, \rho_{0 \max}]$ .

Искомые границы получаются как решение системы неравенств относительно неизвестного  $\rho$  (интервал между минимально допустимым перигеем и максимально допустимым апогеем):

$$(a_{\min}(1-e_{\max}))^2 \leq \|r(\rho)\|^2 \leq (a_{\max}(1+e_{\max}))^2, \rho \geq 0,$$

где  $r(\rho) = R + \rho u, u = (\cos \delta \cos \alpha, \cos \delta \sin \alpha, \sin \delta)^T$ .

**Задача 3.** По данным  $(\alpha, \dot{\alpha}, \delta, \dot{\delta}, R, \dot{R}), a, \rho$  осуществляется построение соответствующих  $\dot{\rho}_{\pm}, e_{\pm}$ .

Искомые значения  $\dot{\rho}_{\pm}$  получаются как решение уравнения относительно  $\dot{\rho}$  (энергия – первый интеграл движения по кеплеровой орбите):

$$\frac{\dot{r}(\dot{\rho}) \cdot \dot{r}(\dot{\rho})}{2} - \frac{\mu}{\|r\|} = -\frac{\mu}{2a},$$

где  $\mu$  – гравитационная постоянная;  $r = R + \rho u, u = (\cos \delta \cos \alpha, \cos \delta \sin \alpha, \sin \delta)^T; \dot{r}(\dot{\rho}) = \dot{R} + \rho \dot{u} + \dot{\rho} u;$

$$\dot{u} = \dot{\delta}(-\sin \delta \cos \alpha, -\sin \delta \sin \alpha, \cos \delta)^T + \dot{\alpha}(-\cos \delta \sin \alpha, \cos \delta \cos \alpha, 0)^T.$$

После чего соответствующие значения  $e_{\pm}$  находятся по формуле (норма вектора Лапласа)

$$e_{\pm} = \left\| \frac{\dot{r}(\rho_{\pm}) \times (r \times \dot{r}(\rho_{\pm}))}{\mu} - \frac{r}{\|r\|} \right\|.$$

**Основная задача**

При заданном значении точности  $\varepsilon$  для сетки узловых значений  $a$  в области  $[a_{\min}, a_{\max}]$  при известных значениях  $(t_1, \alpha_1, \dot{\alpha}_1, \delta_1, \dot{\delta}_1, R_1, \dot{R}_1, t_2, \alpha_2, \dot{\alpha}_2, \delta_2, \dot{\delta}_2, R_2, \dot{R}_2), [e_{\min}, e_{\max}]$  найти границы изменения параметров  $a, \rho, I, \Omega$  возможных орбит.

Сначала решается задача 2 по данным  $(\alpha_1, \delta_1, R_1)$  и  $(\alpha_2, \delta_2, R_2)$  определяется область изменения дальностей  $[\rho_{1min}, \rho_{1max}] \times [\rho_{2min}, \rho_{2max}]$ .

Далее осуществляется перебор узловых значений  $(a, \rho_1, \rho_2)$  по сетке в полученной области  $[a_{min}, a_{max}] \times [\rho_{1min}, \rho_{1max}] \times [\rho_{2min}, \rho_{2max}]$ , а именно: для каждой тройки значений  $(a, \rho_1, \rho_2)$  выполняется следующее.

- Проверяется выполнение условий (наименьший возможный эксцентриситет  $e_0$  и наименьшая возможная длина большой полуоси  $a_0$ , а также теорема Эйлера о времени полета  $\Delta t_p$  между данными точками положения на параболической (с нулевой энергией) орбите):

$$e_0 = \frac{\|r_1\| - \|r_2\|}{\|r_2 - r_1\|} \leq e_{max},$$

$$a_0 = \frac{1}{4} (\|r_1\| + \|r_2\| + \|r_2 - r_1\|) \leq a_{max},$$

$$\Delta t = t_2 - t_1 > \Delta t_p = \frac{4}{3} \sqrt{\frac{a_0^3}{\mu}} (1 - \lambda^3),$$

где  $r_1 = R_1 + \rho_1 u_1, r_2 = R_2 + \rho_2 u_2$ ,

$$0 \leq \lambda^2 = \frac{\|r_1\| + \|r_2\| - \|r_2 - r_1\|}{\|r_1\| + \|r_2\| + \|r_2 - r_1\|} \leq 1.$$

- Вычисляются значения  $I = \arccos(\cos(nk))$ ,  $\Omega = i \cdot \frac{k \times n}{\|k \times n\|}$ , где  $n = \frac{r_1 \times r_2}{\|r_1 \times r_2\|}$ ,  $i, k$  – единичные векторы экваториальной системы координат.

- Решается задача 3 для наборов данных  $(\alpha_1, \dot{\alpha}_1, \delta_1, \dot{\delta}_1, R_1, \dot{R}_1, a, \rho_1)$ ,  $(\alpha_2, \dot{\alpha}_2, \delta_2, \dot{\delta}_2, R_2, \dot{R}_2, a, \rho_2)$ . Получаются значения  $\dot{\rho}_{1\pm}, e_{1\pm}$  и  $\dot{\rho}_{2\pm}, e_{2\pm}$  и их средние величины  $\dot{\rho}_{\pm} = \frac{1}{2}(\dot{\rho}_{1\pm} + \dot{\rho}_{2\pm})$ ,

$e_{\pm} = \frac{1}{2}(e_{1\pm} + e_{2\pm})$ . Вычисляются соответствующие им величины  $I_{\pm}, \Omega_{\pm}$  из соотношений

$$I = \arccos\left(\frac{r \times \dot{r}}{\|r \times \dot{r}\|} k\right), \Omega = \arccos\left(i \cdot \frac{k \times (r \times \dot{r})}{\|k \times (r \times \dot{r})\|}\right).$$

- Проверяется выполнение условий:  $|e_{1\pm} - e_{2\pm}| \leq \varepsilon(e_{max} - e_{min}), |I - I_{\pm}| \leq 180\varepsilon, |\Omega - \Omega_{\pm}| \leq 360\varepsilon$ .

- Вычисляются границы изменения параметров  $a, \rho, I, \Omega$  «хороших» орбит.

Дальнейшее уточнение полученных границ изменения возможных орбит может быть проведено итерационно для уменьшающегося на каждой итерации значения точности  $\varepsilon$ . При этом на каждой последующей итерации в качестве начальных границ изменения дальности выбираются границы, полученные на предыдущей итерации. Соответствующая последовательность полученных границ изменения параметров орбит представляет в этом случае последовательность вло-

женных параллелепипедов в пространстве изменения трех переменных  $(\rho, I, \Omega)$ .

### Вычислительные эксперименты

Входная последовательность включает 13 треков. После выбора в каждом треке двух измерений получены необходимые для вычисления границ параметров орбит данные (табл. 1). Полут ценные границы орбит приведены в таблице 2 (выделены четыре области изменения эксцентриситета:  $e_{area}=1$  соответствует области  $0 \leq e \leq 0,1$ ,  $e_{area}=2-0,1 \leq e \leq 0,3$ ,  $e_{area}=3-0,3 \leq e \leq 0,6$  и  $e_{area}=4-0,6 \leq e \leq 0,9$ ).

На рисунках 1 и 2 в виде проекций параллелепипедов на плоскости  $(I, \Omega)$  и  $(\rho, \Omega)$  соответственно представлены полученные границы орбит для каждого из треков в четвертой области изменения эксцентриситета ( $e_{area}=4$ , то есть  $0,6 \leq e \leq 0,9$ ), так как именно эта область порождает максимальное количество допустимых троек треков.

После анализа пересечений границ параметров возможных орбит по 13 трекам сформировалась последовательность всего из 13 троек треков: 1-5-7, 1-7-8, 2-3-4, 2-3-6, 2-4-6, 3-4-6, 4-6-10, 4-6-11, 4-10-11, 5-6-7, 5-6-12, 5-7-8, 6-10-11.

### Особенности программной реализации

Вычислительные эксперименты направлены на уменьшение средней скорости обработки одного трека при увеличении точности полученной оценки, так как скорость обработки одного трека естественным образом ограничена. Предложено для четырех подобластей изменения эксцентриситета параллельно решать задачу для каждого трека. При этом для каждой подобласти и каждого трека число итераций, необходимых для достижения требуемой точности, заранее неизвестно. Следовательно, при дальнейшей программной реализации алгоритма заранее неизвестно, как будут загружены отдельные узлы вычислительной установки. Поэтому динамические программы предпочтительнее любых других типов.

Динамическое распараллеливание имеет ряд преимуществ не только для случая, когда вопросы организации параллельного счета решаются во время исполнения программы, но и для выравнивания нагрузки в гетерогенных и/или меняющихся со временем параллельных вычислительных системах, а также для задач, обладающих гранулами различной тяжести. При этом использование неявных конструкций распараллеливания вычислений позволяет легко переносить программы между различными платформами параллельных вычислений – многопроцессорными системами, вычислительными кластерами, метаclusterными системами и т.п.

Таблица 1

Входные данные

Table 1

Input data

№	$\Delta t, c$	$\alpha, \text{град}$	$\dot{\alpha}, 10^3 \text{ град/с}$	$\delta, \text{град}$	$\dot{\delta}, 10^3 \text{ град/с}$	$R, \text{ км}$			$\dot{R}, \text{ км/с}$		
1	70	5,129	0,073	-0,370	0,008	-163,397	-4621,021	4381,470	0,336	-0,011	0
		5,134	0,071	-0,369	0,008	-139,807	-4621,795	4381,470	0,337	-0,010	0
2	254	3,908	0,071	-0,086	-0,012	435,094	-4603,393	4381,470	0,335	0,031	0
		3,926	0,071	-0,089	-0,012	520,278	-4594,545	4381,470	0,335	0,037	0
3	82	3,924	0,064	-0,091	-0,013	515,252	-4595,111	4381,470	0,335	0,037	0
		3,929	0,063	-0,092	-0,012	542,720	-4591,948	4381,470	0,334	0,039	0
4	142	4,529	0,067	-0,260	-0,013	-37,999	-4623,753	4381,470	0,337	-0,002	0
		4,539	0,067	-0,262	-0,013	9,879	-4623,898	4381,470	0,337	0,000	0
5	344	5,027	0,075	-0,318	0,011	-728,199	-4566,208	4381,470	0,332	-0,053	0
		5,052	0,075	-0,314	0,011	-613,439	-4583,037	4381,470	0,334	-0,044	0
6	508	3,544	0,086	-0,257	-0,006	-2099,929	-4119,567	4381,470	0,300	-0,153	0
		3,588	0,086	-0,260	-0,006	-1945,918	-4194,512	4381,470	0,305	-0,141	0
7	494	5,015	0,083	-0,341	0,009	-434,892	-4603,412	4381,470	0,335	-0,031	0
		5,056	0,083	-0,336	0,010	-268,817	-4616,088	4381,470	0,336	-0,019	0
8	157	4,605	0,057	-0,408	0,000	-2628,813	-3803,929	4381,470	0,277	-0,191	0
		4,614	0,057	-0,408	0,000	-2585,092	-3833,775	4381,470	0,279	-0,188	0
9	224	4,530	0,095	-0,001	0,006	-2381,782	-3963,288	4381,470	0,289	-0,173	0
		4,551	0,094	0,000	0,005	-2316,729	-4001,662	4381,470	0,291	-0,168	0
10	538	4,012	0,055	-0,293	-0,012	-1431,269	-4396,817	4381,470	0,320	-0,104	0
		4,042	0,059	-0,300	-0,013	-1257,718	-4449,571	4381,470	0,324	-0,091	0
11	67	3,945	0,072	-0,287	-0,016	-1706,203	-4297,604	4381,470	0,313	-0,124	0
		3,950	0,072	-0,289	-0,016	-1685,186	-4305,889	4381,470	0,313	-0,122	0
12	426	4,622	0,071	-0,183	0,006	-3705,851	-2765,357	4381,470	0,201	-0,270	0
		4,653	0,072	-0,181	0,006	-3618,173	-2879,125	4381,470	0,209	-0,263	0
13	456	4,045	0,056	-0,080	-0,002	-3370,538	-3165,439	4381,470	0,230	-0,245	0
		4,071	0,055	-0,081	-0,002	-3263,437	-3275,746	4381,470	0,238	-0,237	0

T-система (OpenTS) – система параллельного программирования, реализующая концепцию автоматического динамического распараллеливания

программ. Это оригинальная российская разработка, которая ведется в Институте программных систем им. А.К. Айламазяна РАН [10, 11], пред-

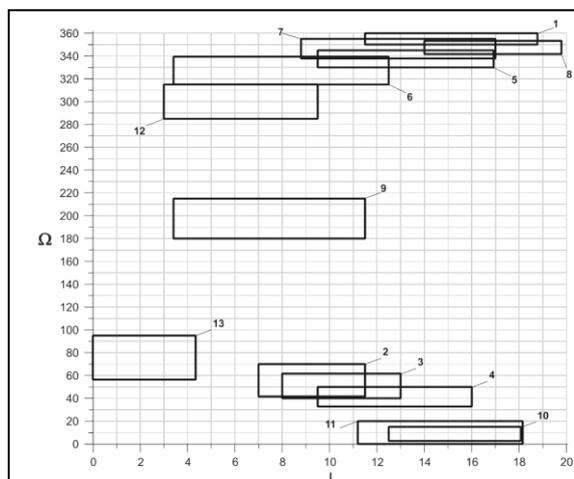


Рис. 1. Полученные границы орбит в проекции на плоскость (l,  $\Omega$ )  
 Fig. 1. The obtained boundaries of orbits projected onto the plane (l,  $\Omega$ )

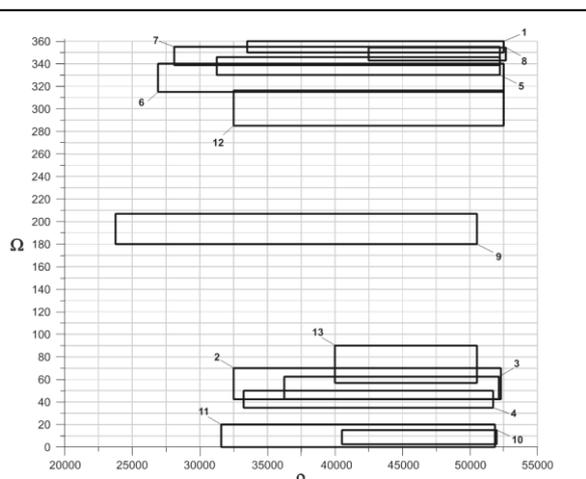


Рис. 2. Полученные границы орбит в проекции на плоскость (rho,  $\Omega$ )  
 Fig. 2. The obtained boundaries of orbits projected onto the plane (rho,  $\Omega$ )

ставляющая систему параллельного программирования, реализующую концепцию автоматического динамического распараллеливания программ.

То есть многие аспекты организации параллельного счета (распределение задач по узлам кластера, синхронизация процессов, организация

Таблица 2

## Полученные границы орбит

Table 2

## The obtained boundaries of orbits

№	$e_{area}$	$\rho_{min}$ КМ	$\rho_{max}$ КМ	$I_{min}$ Град	$I_{max}$ Град	$\Omega_{min}$ Град	$\Omega_{max}$ Град
1	1	33613,956	52381,905	11,502	18,826	350,083	360,000
	2	37868,200	41224,697	10,231	19,778	345,737	360,000
	3	30605,722	39128,835	10,962	17,196	346,360	359,714
	4	24032,107	33537,704	9,053	16,140	339,655	357,212
2	1	32916,409	52214,004	6,921	11,529	42,424	69,222
	2	36198,715	41244,384	7,475	12,124	50,406	64,303
	3	30538,450	39042,829	7,495	12,485	52,543	71,444
	4	24070,227	35733,542	7,607	13,340	56,040	81,844
3	1	36364,279	52228,56	8,139	12,881	42,021	62,960
	2	37752,279	39863,02	8,835	13,376	50,425	61,222
	3	32145,486	39050,28	8,848	13,590	51,042	67,056
	4	21110,193	36293,45	8,917	15,029	53,626	84,111
4	1	33787,297	51468,022	9,517	16,023	33,918	50,967
	2	37197,046	40519,339	10,113	15,026	38,298	49,517
	3	31951,552	40190,632	9,350	14,987	38,466	53,257
	4	22301,983	35387,762	7,603	14,367	41,413	67,728
5	1	31468,991	52125,664	9,679	17,121	329,863	347,046
	2	34632,070	37979,655	8,254	17,785	326,070	346,162
	3	30503,026	38997,540	9,740	15,683	327,086	342,204
	4	15239,623	33442,988	6,360	14,798	285,263	338,688
6	1	27860,800	52359,886	4,044	12,366	314,438	338,537
	2	31750,087	35105,842	5,250	10,349	325,057	337,961
	3	31736,272	35244,535	5,267	10,372	324,922	337,844
	4	16663,041	31619,544	3,095	9,725	31,747	338,143
7	1	28584,758	52123,370	8,786	17,226	338,882	356,068
	2	34630,436	37977,939	3,657	22,206	327,380	360,000
	3	26641,080	35135,337	8,488	14,975	333,942	350,078
	4	20402,859	33672,775	6,507	14,687	322,075	349,218
8	1	43013,267	52653,863	14,005	19,739	344,786	354,426
	3	36213,131	41545,179	12,793	18,259	342,141	352,266
	4	27876,113	38045,739	10,566	17,645	338,940	351,567
9	1	23984,972	50481,738	3,250	11,393	181,259	207,403
	3	22950,351	29930,643	3,490	14,151	178,859	200,918
	4	15946,175	24319,452	7,071	14,881	178,828	193,045
10	1	41193,199	51916,879	12,488	18,126	3,457	15,718
	4	22780,690	27843,452	3,408	22,390	5,815	48,807
11	1	32196,152	51851,370	11,273	18,164	1,241	18,866
	3	30392,904	37333,259	9,044	19,052	3,055	26,256
	4	18451,387	33571,305	9,068	16,299	9,988	43,265
12	1	33167,207	52421,258	2,966	9,382	286,612	315,303
	3	30621,479	38488,098	1,283	10,806	271,593	304,872
	4	22945,746	33089,442	3,329	8,176	249,252	291,019
13	1	40476,340	50585,775	0,000	4,534	56,637	93,813
	3	33664,725	41986,343	0,000	7,469	78,203	117,547
	4	22659,360	38082,958	0,170	7,722	93,530	130,877

пересылки данных) выполняются не программистом, а самой системой. Во многих случаях Т-системе удастся удачно организовать все эти виды работ и получить хороший уровень распараллеливания программ.

Язык Т++ является входным языком Т-системы (OpenTS). Синтаксически язык Т++ максимально приближен к языку С++.

В языке Т++ поддерживается функциональный подход к написанию программ: все данные Т-функция («чистая» функция в языке Т++) может получать только через входные аргументы, а результаты своей работы возвращать только через выходные аргументы. Т-функция не должна иметь побочных эффектов. Таким образом, Т-функция является гранулой параллелизма: при вызове данной функции всю информацию она получает через свои аргументы, поэтому ее можно отдать на выполнение другому процессору.

Описанный выше алгоритм решения задачи априорного определения границ изменения параметров возможных орбит реализован в Т-системе (язык программирования Т++). Рекомендуется следующая организация разработки программы на языке Т++.

1. На этапе построения алгоритма разрабатывается его верхний уровень на базе парадигмы функционального программирования.

2. Этап разработки программного кода включает в себя решение вопроса о том, какие фрагменты алгоритма (какая часть программного кода) будут реализованы в виде Т-функций, а какая часть в виде привычного последовательного исполняемого кода в стандарте языка С++.

3. Этап реализации программы и ее первичной отладки на однопроцессорном компьютере.

4. Этап отладки на многопроцессорной установке заключается в отладке программы на одиночном SMP-компьютере и в дальнейшем запуске на кластере.

Отметим, что при реализации программы для Т-системы программист обязан адекватно изложить алгоритм в функциональном стиле – описать программу в виде набора Т-функций. Кроме этого, он должен стремиться выбрать оптимальный размер Т-функции (оптимально подобрать среднюю вычислительную сложность Т-функции). Так, слишком малая вычислительная сложность Т-функций может привести к большим накладным расходам (например, время, затраченное на передачу Т-функций и данных для них на другие узлы кластера, превысит время счета самой этой функции). В то же время слишком большая вычислительная сложность Т-функций может привести к малому количеству порождаемых в процессе счета гранул параллелизма и, как следствие, к неравномерной загрузке вычислительных узлов кластера. Рассмотрим фрагмент кода, отвечающий за параллельное исполнение программы:

```
#define ND 100
#define NBOUNDS 64
#define NVEC 26
...
struct result_bounds_s {
    double result_bounds[NBOUNDS];
};

struct vnod_vector_s {
    double vnod_vector[NVEC];
};
...
tfun struct result_bounds_s track_bounds(unsigned int Max_iters,
                                        double eps_const,
                                        struct vnod_vector_s
                                        vnod_vector_e) {

// код Т-функции
...
}

tfun int main(int argc, char* argv[]) {
    ...
    struct vnod_vector_s vnod_vector_e;
    ...
    tval struct result_bounds_s result_bounds_t[ND];
    for (unsigned int data = 0; data < ND; data++) {
        ...
        result_bounds_t[ND] = track_bounds(Max_iters,
            eps_const, vnod_vector_e);
        ...
    }
    ...
    return 0;
}
...

```

Видно, что при написании программы к стандартному набору С++ были добавлены лишь два ключевых слова языка Т++: `tval` и `tfun`.

Атрибут `tval` можно указывать непосредственно перед описанием типа переменной, при этом переменная кратко называется Т-переменной и содержит неготовую величину. Неготовые величины производятся с помощью вызова Т-функций и в каждый момент времени либо не готовы (при этом их значение не определено, а попытка обращения к ним влечет за собой приостановку обрабатываемой функции), либо готовы, то есть уже посчитаны. Нельзя изменить собственно неготовую величину, но можно присвоить Т-переменной другую неготовую величину.

Атрибут `tfun` указывается непосредственно перед описанием типа функции, а соответствующая функция кратко называется Т-функцией (в данном случае Т-функция – это функция `track_bounds()`). Т-функции являются гранулами параллелизма. При этом они могут одновременно выполняться на разных процессорах (узлах) в многопроцессорной системе. Отметим, что функция `main()` также должна быть объявлена как Т-функция.

Описанный выше алгоритм решения задачи априорного определения границ изменения параметров возможных орбит реализован в Т-системе (язык программирования Т++). Результаты исследования эффективности параллельной версии программы для обработки 100 треков представлены в таблице 3. При расчете в последовательном режи-

ме, исходя из требований целостности процесса работы комплекса программ ОТИ, наложено ограничение на среднее время обработки одного трека значением 0,7 с. Данное ограничение позволяет провести лишь три итерации и, как следствие, достигнуть относительной погрешности 2,5 %.

Задача состояла в увеличении числа итераций (для достижения требуемой точности 0,3 %) с сохранением среднего времени обработки одного трека.

Таблица 3

**Время выполнения  
параллельной версии программы**

Table 3

**Execution time of the parallel version of the program**

Число ядер	Время, с	Среднее время обработки одного трека, с
1	195,178	1,95
2	146,233	1,46
4	86,399	0,86
8	55,220	0,55
16	53,346	0,53

Из таблицы видно, что для достижения выбранной точности среднее время обработки одного трека удается сделать приемлемым при расчете на 8 ядрах суперкомпьютера. Все эксперименты производились на тестовом кластере, расположенном в Институте программных систем имени А.К. Айламазяна РАН.

На основании изложенного можно сделать следующие выводы. Переход к регулярному наблюдению и каталогизации малоразмерных ОКМ потребует совершенствования применяемых методов и алгоритмов получения и обработки информации. Одним из направлений решения этой проблемы является повышение эффективности работы оптических средств, которым в мониторинге малоразмерных ОКМ отводится значительная роль. В статье для ускорения обнаружения и определения орбит некаталогизированных ОКМ предложен вспомогательный алгоритм построения для каждого трека границ параметров возможных орбит. Подобный подход позволяет оптимизировать работу комплекса программ ОТИ в целом и, в частности, существенно ускорить работу по определению орбит новых КО непосредственно в процессе сеанса наблюдений.

Динамическое распараллеливание, обеспечиваемое Т-системой, имеет ряд преимуществ не только для случая, когда вопросы организации па-

раллельного счета решаются во время исполнения программы, но и для выравнивания нагрузки в гетерогенных и/или меняющихся со временем параллельных вычислительных системах, а также для задач, обладающих гранулами различной тяжести, как, например, для случая реализованного алгоритма априорного расчета границ параметров возможных орбит.

**Литература**

1. Yurasov V.S., Vygon V.G., Shargorodsky V.D. Features of detection and orbit determination of uncatalogued space debris // Proceedings of The Seventh US-Russian Space Surveillance Workshop, 2007. URL: <http://aero.tamu.edu/sites/default/files/faculty/alfriend/R7%20S3.1%20Yurasov.pdf> (дата обращения: 19.06.2015).
2. Alby F. et al. Status of CNES optical observations of space debris in geostationary orbit // Advances in Space Research, 2004, vol. 34, no. 5, pp. 1143–1149.
3. Earl M.A., Racey T.J. The Canadian Automatic Small Telescope for Orbital Research (CASTOR). A RAVEN System in Canada // 1999 AMOS Technical Conference, Aug., 1999, pp. 401–410.
4. Klinkrad H. Monitoring Space—Efforts Made by European Countries // International Colloquium on Europe and Space Debris, sponsored by the Academie National de l’Air et de l’Espace, Toulouse, France (27–28 November 2002). URL: <http://www.fas.org/spp/military/program/track/klinkrad.pdf> (дата обращения: 19.06.2015).
5. Flury W. et al. Searching for small debris in the geostationary ring // ESA bulletin, 2000, vol. 104, pp. 92–100.
6. Payne P.T. New Deep Space Optical Search Strategies. Proceedings of the Fifth US-Russian Space Surveillance Workshop. St. Petersburg, 2003. URL: <http://aero.tamu.edu/sites/default/files/faculty/alfriend/Russia6thWorkshop/r5%20S5.3%20Payne.pdf> (дата обращения: 19.06.2015).
7. Schildknecht T. et al. An optical search for small-size debris in GEO and GTO. Proceedings of the 2003 AMOS Technical Conference, 2003, vol. 9, pp. 12.
8. Flohrer T., Schildknecht T., Musci R. Proposed strategies for optical observations in a future European Space Surveillance network. Advances in Space Research, 2008, vol. 41, no. 7, pp. 1010–1021.
9. Roscoe C.W.T., Schumacher Jr P.W., Wilkins M.P. Parallel Track Initiation for Optical Space Surveillance using Range and Range-Rate Bounds. Advances in the Astronautical Sciences, 2014, vol. 150, pp. 989–1008.
10. Абрамов С.М., Кузнецов А.А., Роганов В.А. Кросс-платформенная версия Т-системы с открытой архитектурой // Параллельные вычислительные технологии (ПаВТ’2007): тр. Междунар. науч. конф. Челябинск: Изд-во ЮУрГУ. 2007. Т. 1. С. 115–121.
11. Кузнецов А.А., Роганов В.А. Экспериментальная реализация отказоустойчивой версии системы OPENTS для платформы WINDOWS CCS // Суперкомпьютерные системы и их применение (SSA’2008): тр. 2-й Междунар. науч. конф. Минск: Изд-во ОИПИ НАН Беларуси, 2008. С. 65–70.

DOI: 10.15827/0236-235X.111.080-087

Received 19.07.15

**OPTIMIZATION FOR THE DETECTION PROCESS OF NEW SPACE OBJECT ORBITS  
BY A PARALLEL CALCULATING OF POSSIBLE ORBITS**

**Trushkova E.A.**, Dr.Sc. (Physics and Mathematics), Leading Research, katerinatr@mail.ru  
(V.A. Trapeznikov Institute of Control Sciences of RAS, Profsoyuznaya St. 65, 117997, Russian Federation);  
**Matveev G.A.**, Leading Research Engineer, gera@prime.botik.ru

(Program System Institute of RAS, Petr I Ave. 4a, Pereslavl-Zalessky, 152021, Russian Federation)

**Abstract.** Observation and cataloging of small debris objects in near–Earth space requires the improvement of the methods and algorithms for obtaining and processing information to improve the efficiency of the optical means. One of the approaches for operative detection of uncatalogued space debris objects orbits has been tested and used successfully in Russian optical observation ground facilities. However, much time is spent on the operation of the exhaustive search of short series of measurements (tracks) and the choice of three series separated in time, presumably belonging to the same object debris and used for the subsequent construction of a possible initial orbit.

The article considers the issue of reducing tracks exhaustive search using preliminary analysis of the measurements by a priori constructing the boundaries of possible orbits parameters' changes. Computational experiments within the research of this application problem are aimed at reducing the average processing speed of one track by increasing the accuracy of estimation. The processing speed of one track is naturally limited since the construction of the possible orbits parameter estimates for each track should not greatly affect the time of the overall measurement processing cycle as part of a trajectory measurements general processing program. The decrease in the average processing speed of one track and the simultaneous increasing the accuracy of estimates is possible with a parallel operation of the program. So, in order to optimize the program for detecting and determining the orbits of new space objects it is proposed to use a parallel algorithm for constructing the boundaries of the possible orbits changes for selection of allowable triples of tracks which have nonempty intersection. The algorithm is implemented in T++ for a T-system with an open architecture (OpenTS).

**Keywords:** space debris, near–Earth space, the initial orbit, parallel algorithm, dynamic parallelization, T++ programming language, OpenTS, T-system, C++ parallel extension.

### References

1. Yurasov V.S., Vygon V.G., Shargorodsky V.D. Features of detection and orbit determination of uncatalogued space debris. *Proc. of the 7th US-Russian Space Surveillance Workshop*. 2007. Available at: <http://aero.tamu.edu/sites/default/files/faculty/alfriend/R7%20S3.1%20Yurasov.pdf> (accessed June 19, 2015).
2. Alby F. Status of CNES optical observations of space debris in geostationary orbit. *Advances in Space Research*. 2004, vol. 34, no. 5, pp. 1143–1149.
3. Earl M.A., Racey T.J. The Canadian Automatic Small Telescope for Orbital Research (CASTOR) – A RAVEN System in Canada. *1999 AMOS Technical Conf*. 1999, pp. 401–410.
4. Klinkrad H. Monitoring Space–Efforts Made by European Countries. *Int. Colloquium on Europe and Space Debris, sponsored by the Academie National de l’Air et de l’Espace*. Toulouse, France, 2002, Federation of American Scientists. Available at: <http://www.fas.org/spp/military/program/track/klinkrad.pdf> (accessed June 19, 2015).
5. Flury W. Searching for small debris in the geostationary ring. *ESA bulletin*. 2000, vol. 104, pp. 92–100.
6. Payne P.T. New Deep Space Optical Search Strategies. *Proc. of the 5th US-Russian Space Surveillance Workshop*. St. Petersburg, 2003. Available at: <http://aero.tamu.edu/sites/default/files/faculty/alfriend/Russia6th-Workshop/r5%20S5.3%20Payne.pdf> (accessed June 19, 2015).
7. Schildknecht T. An optical search for small-size debris in GEO and GTO. *Proc. of the 2003 AMOS Technical Conf*. 2003, vol. 9, pp. 12.
8. Flohrer T., Schildknecht T., Musci R. Proposed strategies for optical observations in a future European Space Surveillance network. *Advances in Space Research*. 2008, vol. 41, no. 7, pp. 1010–1021.
9. Roscoe C.W.T., Schumacher Jr. P.W., Wilkins M.P. Parallel Track Initiation for Optical Space Surveillance using Range and Range-Rate Bounds. *Advances in the Astronautical Sciences*. 2014, vol. 150, pp. 989–1008.
10. Abramov S.M., Kuznetsov A.A., Roganov V.A. T-system cross-platform version with open architecture. *Trudy Mezhdunar. nauch. konf. “Parallelnye vychislitelnye tekhnologii (PAVT'2007)”* [Proc. of the Int. Scientific Conf. “Parallel Computing Technologies PCT'2007”]. Chelyabinsk, 2007, vol. 1, pp. 115–121 (in Russ.).
11. Kuznetsov A.A., Roganov V.A. An experimental implementation of OPENTS system fault-tolerant version for WINDOWS CCS. *Trudy 2 Mezhdunar. nauch. konf. “Superkomputernye systemy i ikh primeneniye (SSA'2008)”* [Proc. of the 2nd Int. Scientific Conf. “Supercomputer Systems and Applications (SSA'2008)”]. Minsk, OIPI NAN Belarusi Publ., 2008, pp. 65–70 (in Russ.).