

DRAGONFLY

Precise 3D location for robots and drones using standard cameras

Dragonfly is a visual positioning system that provides precise location in 3 dimensions to automated equipment, such as autonomous mobile robots and drones.

Applying patented simultaneous localization and mapping (**SLAM**) technology, Dragonfly uses the equipment's camera to deliver highly accurate location. No other sensor is needed.

Dragonfly helps solve a major problem: GPS does not work indoors, and it may not be accurate enough outdoors. Robots, Drones, Automated and Unmanned Vehicles cannot rely on GPS for most of the operations.

Dragonfly provides instead a reliable, effective and accurate location, by using only an on-board camera. The camera is the only sensor required to compute location. Dragonfly does simultaneous localization and mapping (SLAM), performing visual odometry while simultaneously mapping the surrounding environment, locating the mobile device in that reconstructed environment.

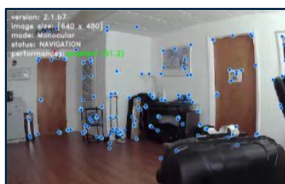
FEATURES

Dragonfly computes the device's location in a venue and makes it available for external applications.



Coordinates

The location is provided as WSG-84 (latitude, longitude + and altitude off the floor) and Cartesian Coordinates (x, y, z).



Physical working environment

Dragonfly can work indoors, outdoors or a mix of both.

Positioning Status	
Navigation	
Markers	Map Points
3	4264
Longitude	Altitude[m]
-80.130512	1.28

Accuracy

Dragonfly can deliver an average location accuracy better than 5 cm, when properly calibrated.

```

1 // initialize = 1
2 // test = 4
3
4 public int getState() {return mState;}
5
6 //
7 // Return current latitude
8 //
9 public double getLatitude() {return mStateHolder.getLatitude();}
10
11 //
12 // Return current longitude
13 //
14 public double getLongitude() {return mStateHolder.getLongitude();}
15
16 //
17 // Return current altitude
18 //
19 public double getAltitude() {return mStateHolder.getAltitude();}
20
21 //
22 //
23 //
24 //
25 //
26 //
27 //
28 //
29 //
30 //
31 //
32 //
33 //
34 //
35 //
36 //
37 //
38 //
39 //
40 //
41 //
42 //
43 //
44 //
45 //
46 //
47 //
48 //
49 //
50 //
51 //
52 //
53 //
54 //
55 //
56 //
57 //
58 //
59 //
60 //
61 //
62 //
63 //
64 //
65 //
66 //
67 //
68 //
69 //
70 //
71 //
72 //
73 //
74 //
75 //
76 //
77 //
78 //
79 //
80 //
81 //
82 //
83 //
84 //
85 //
86 //
87 //
88 //
89 //
90 //
91 //
92 //
93 //
94 //
95 //
96 //
97 //
98 //
99 //
100 //
101 //
102 //
103 //
104 //
105 //
106 //
107 //
108 //
109 //
110 //
111 //
112 //
113 //
114 //
115 //
116 //
117 //
118 //
119 //
120 //
121 //
122 //
123 //
124 //
125 //
126 //
127 //
128 //
129 //
130 //
131 //
132 //
133 //
134 //
135 //
136 //
137 //
138 //
139 //
140 //
141 //
142 //
143 //
144 //
145 //
146 //
147 //
148 //
149 //
150 //
151 //
152 //
153 //
154 //
155 //
156 //
157 //
158 //
159 //
160 //
161 //
162 //
163 //
164 //
165 //
166 //
167 //
168 //
169 //
170 //
171 //
172 //
173 //
174 //
175 //
176 //
177 //
178 //
179 //
180 //
181 //
182 //
183 //
184 //
185 //
186 //
187 //
188 //
189 //
190 //
191 //
192 //
193 //
194 //
195 //
196 //
197 //
198 //
199 //
200 //
201 //
202 //
203 //
204 //
205 //
206 //
207 //
208 //
209 //
210 //
211 //
212 //
213 //
214 //
215 //
216 //
217 //
218 //
219 //
220 //
221 //
222 //
223 //
224 //
225 //
226 //
227 //
228 //
229 //
230 //
231 //
232 //
233 //
234 //
235 //
236 //
237 //
238 //
239 //
240 //
241 //
242 //
243 //
244 //
245 //
246 //
247 //
248 //
249 //
250 //
251 //
252 //
253 //
254 //
255 //
256 //
257 //
258 //
259 //
260 //
261 //
262 //
263 //
264 //
265 //
266 //
267 //
268 //
269 //
270 //
271 //
272 //
273 //
274 //
275 //
276 //
277 //
278 //
279 //
280 //
281 //
282 //
283 //
284 //
285 //
286 //
287 //
288 //
289 //
290 //
291 //
292 //
293 //
294 //
295 //
296 //
297 //
298 //
299 //
300 //
301 //
302 //
303 //
304 //
305 //
306 //
307 //
308 //
309 //
310 //
311 //
312 //
313 //
314 //
315 //
316 //
317 //
318 //
319 //
320 //
321 //
322 //
323 //
324 //
325 //
326 //
327 //
328 //
329 //
330 //
331 //
332 //
333 //
334 //
335 //
336 //
337 //
338 //
339 //
340 //
341 //
342 //
343 //
344 //
345 //
346 //
347 //
348 //
349 //
350 //
351 //
352 //
353 //
354 //
355 //
356 //
357 //
358 //
359 //
360 //
361 //
362 //
363 //
364 //
365 //
366 //
367 //
368 //
369 //
370 //
371 //
372 //
373 //
374 //
375 //
376 //
377 //
378 //
379 //
380 //
381 //
382 //
383 //
384 //
385 //
386 //
387 //
388 //
389 //
390 //
391 //
392 //
393 //
394 //
395 //
396 //
397 //
398 //
399 //
400 //
401 //
402 //
403 //
404 //
405 //
406 //
407 //
408 //
409 //
410 //
411 //
412 //
413 //
414 //
415 //
416 //
417 //
418 //
419 //
420 //
421 //
422 //
423 //
424 //
425 //
426 //
427 //
428 //
429 //
430 //
431 //
432 //
433 //
434 //
435 //
436 //
437 //
438 //
439 //
440 //
441 //
442 //
443 //
444 //
445 //
446 //
447 //
448 //
449 //
450 //
451 //
452 //
453 //
454 //
455 //
456 //
457 //
458 //
459 //
460 //
461 //
462 //
463 //
464 //
465 //
466 //
467 //
468 //
469 //
470 //
471 //
472 //
473 //
474 //
475 //
476 //
477 //
478 //
479 //
480 //
481 //
482 //
483 //
484 //
485 //
486 //
487 //
488 //
489 //
490 //
491 //
492 //
493 //
494 //
495 //
496 //
497 //
498 //
499 //
500 //
501 //
502 //
503 //
504 //
505 //
506 //
507 //
508 //
509 //
510 //
511 //
512 //
513 //
514 //
515 //
516 //
517 //
518 //
519 //
520 //
521 //
522 //
523 //
524 //
525 //
526 //
527 //
528 //
529 //
530 //
531 //
532 //
533 //
534 //
535 //
536 //
537 //
538 //
539 //
540 //
541 //
542 //
543 //
544 //
545 //
546 //
547 //
548 //
549 //
550 //
551 //
552 //
553 //
554 //
555 //
556 //
557 //
558 //
559 //
560 //
561 //
562 //
563 //
564 //
565 //
566 //
567 //
568 //
569 //
570 //
571 //
572 //
573 //
574 //
575 //
576 //
577 //
578 //
579 //
580 //
581 //
582 //
583 //
584 //
585 //
586 //
587 //
588 //
589 //
590 //
591 //
592 //
593 //
594 //
595 //
596 //
597 //
598 //
599 //
600 //
601 //
602 //
603 //
604 //
605 //
606 //
607 //
608 //
609 //
610 //
611 //
612 //
613 //
614 //
615 //
616 //
617 //
618 //
619 //
620 //
621 //
622 //
623 //
624 //
625 //
626 //
627 //
628 //
629 //
630 //
631 //
632 //
633 //
634 //
635 //
636 //
637 //
638 //
639 //
640 //
641 //
642 //
643 //
644 //
645 //
646 //
647 //
648 //
649 //
650 //
651 //
652 //
653 //
654 //
655 //
656 //
657 //
658 //
659 //
660 //
661 //
662 //
663 //
664 //
665 //
666 //
667 //
668 //
669 //
670 //
671 //
672 //
673 //
674 //
675 //
676 //
677 //
678 //
679 //
680 //
681 //
682 //
683 //
684 //
685 //
686 //
687 //
688 //
689 //
690 //
691 //
692 //
693 //
694 //
695 //
696 //
697 //
698 //
699 //
700 //
701 //
702 //
703 //
704 //
705 //
706 //
707 //
708 //
709 //
710 //
711 //
712 //
713 //
714 //
715 //
716 //
717 //
718 //
719 //
720 //
721 //
722 //
723 //
724 //
725 //
726 //
727 //
728 //
729 //
730 //
731 //
732 //
733 //
734 //
735 //
736 //
737 //
738 //
739 //
740 //
741 //
742 //
743 //
744 //
745 //
746 //
747 //
748 //
749 //
750 //
751 //
752 //
753 //
754 //
755 //
756 //
757 //
758 //
759 //
760 //
761 //
762 //
763 //
764 //
765 //
766 //
767 //
768 //
769 //
770 //
771 //
772 //
773 //
774 //
775 //
776 //
777 //
778 //
779 //
780 //
781 //
782 //
783 //
784 //
785 //
786 //
787 //
788 //
789 //
790 //
791 //
792 //
793 //
794 //
795 //
796 //
797 //
798 //
799 //
800 //
801 //
802 //
803 //
804 //
805 //
806 //
807 //
808 //
809 //
810 //
811 //
812 //
813 //
814 //
815 //
816 //
817 //
818 //
819 //
820 //
821 //
822 //
823 //
824 //
825 //
826 //
827 //
828 //
829 //
830 //
831 //
832 //
833 //
834 //
835 //
836 //
837 //
838 //
839 //
840 //
841 //
842 //
843 //
844 //
845 //
846 //
847 //
848 //
849 //
850 //
851 //
852 //
853 //
854 //
855 //
856 //
857 //
858 //
859 //
860 //
861 //
862 //
863 //
864 //
865 //
866 //
867 //
868 //
869 //
870 //
871 //
872 //
873 //
874 //
875 //
876 //
877 //
878 //
879 //
880 //
881 //
882 //
883 //
884 //
885 //
886 //
887 //
888 //
889 //
890 //
891 //
892 //
893 //
894 //
895 //
896 //
897 //
898 //
899 //
900 //
901 //
902 //
903 //
904 //
905 //
906 //
907 //
908 //
909 //
910 //
911 //
912 //
913 //
914 //
915 //
916 //
917 //
918 //
919 //
920 //
921 //
922 //
923 //
924 //
925 //
926 //
927 //
928 //
929 //
930 //
931 //
932 //
933 //
934 //
935 //
936 //
937 //
938 //
939 //
940 //
941 //
942 //
943 //
944 //
945 //
946 //
947 //
948 //
949 //
950 //
951 //
952 //
953 //
954 //
955 //
956 //
957 //
958 //
959 //
960 //
961 //
962 //
963 //
964 //
965 //
966 //
967 //
968 //
969 //
970 //
971 //
972 //
973 //
974 //
975 //
976 //
977 //
978 //
979 //
980 //
981 //
982 //
983 //
984 //
985 //
986 //
987 //
988 //
989 //
990 //
991 //
992 //
993 //
994 //
995 //
996 //
997 //
998 //
999 //
1000 //
    
```

Programming environment





Dragonfly comes as a Java application with full support for Linux and MacOS. Different integrations and environments can be provided upon request.

Positioning	
Function provides means to start and stop a positioning session. By using <code>start()</code> you advise the software to no longer start positioning.	
Start positioning	
Request-Resource:	PJT API/1/positioning/start
URL:	localhost:8080/api/1/positioning/start -> API
URL:	localhost:8080/api/1/positioning/start?token=abc123 -> API
Request-Parameter:	Query-String
Parameter	Meaning
Content	How to access simultaneous mapping
Response:	200 OK, application/json

Integration

Dragonfly provides API for custom integration into external applications.

SYSTEM COMPONENT

<p>Camera</p> <p>One camera (mono or stereo) has to be mounted on board of the devices. Dragonfly can leverage the existing cameras, if any.</p>	
<p>Server</p> <p>Dragonfly runs on a server that collects the video streams from the camera. The server can be an on-board machine, or a remote server to which the devices connect over an internet connection.</p>	
<p>Dashboard</p> <p>The dashboard allows the management of the site's data and integrates the visualization of real time positions.</p>	
<p>Markers</p> <p>Visual Markers are graphical images that identify a specific 3D location. Three of them are deployed through the venue for the initial mapping calibration*. They can be removed after the first calibration.</p>	

SYSTEM CALIBRATION

Dragonfly must be calibrated prior to using the system, to provide a valid geo-reference. The calibration process establishes the relationship between the map and global coordinates (latitude and longitude, or x, y, z), as well as altitude off the floor.

The calibration is performed by using at least three Visual Markers*:

- **Visual Markers** are QR code-like patterns that encode their 3D coordinates in the real world: latitude, longitude and altitude.
- **The camera estimates its 3D position** relative to the marker.
- **Markers are generated through Dragonfly's dashboard**, and are printed on regular printers.
- **Markers can be removed** once the calibration is complete

** Visual Markers usage is optional. Dragonfly also provides a feature which allows to perform a manual calibration without visual marker by setting reference points on the floor plan while navigating.*

EQUIPEMENT

Monocular or Stereo Cameras

Dragonfly can work with monocular and stereo cameras: there are important differences to consider when choosing the type of camera.

Type of Camera	Monocular Camera	Stereo Camera
Accuracy	5-10 cm	5-10 cm
Server	Intel Core i5 Quad-core,16GB RAM	Intel core i7 Quad-core,32GB RAM
Pure rotation movements*	NO	YES
Resolution	VGA (640x480 px resolution)	VGA (640x480 px resolution)
Field of View	120-170°	90-120° (Each camera)

** Pure Rotation happens when the camera rotates on just one axis. This happens frequently on drones, for example. Dragonfly cannot support this movement on monocular cameras, unless a preliminary "site survey" has been performed.*

Server

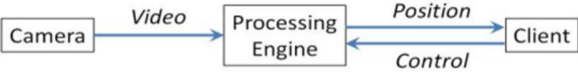
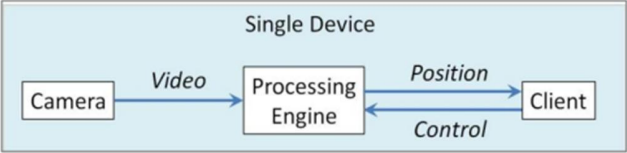
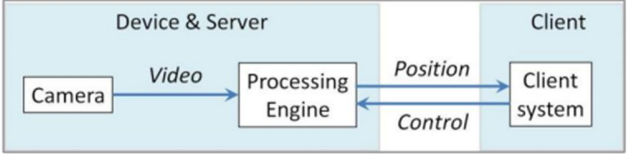
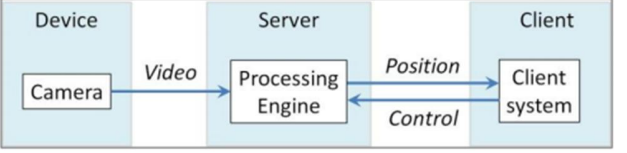
Dragonfly requires a powerful computing unit, where the location engine runs. The server can be mounted on board of the device, or it can be a remote server. These are the servers' minimum requirements for a single monocular camera.

CPU	Real Quad-core*, at least <i>AMD Ryzen 3</i> or <i>Intel Core i5-8250U</i>
RAM	16 GB
O/S	At least Ubuntu 16.04 or MacOS High Sierra

** Real quad-core means that the CPU needs to have at least 4 physical cores. A CPU with 4 simulated cores such as a 2 cores CPU with 4 threads is not recommend.*

SYSTEM ARCHITECTURE

Dragonfly is provided in different configurations to accommodate a variety of needs.

Version 1	The basic system architecture consists of a camera streaming digital video to a processing engine, which forwards the device's position to a client capable of issuing controls.	 <pre> graph LR Camera[Camera] -- Video --> PE[Processing Engine] PE -- Position --> Client[Client] Client -- Control --> PE </pre>
Version 2	All 3 components may run on one single device, that includes the camera, the processing unit, and the client to visualize the location.	 <pre> graph LR subgraph Single_Device [Single Device] Camera[Camera] -- Video --> PE[Processing Engine] PE -- Position --> Client[Client] Client -- Control --> PE end </pre>
Version 3	Camera (USB) and processing engine may run on the same device, communicating with a remote client via the API.	 <pre> graph LR subgraph Device_Server [Device & Server] Camera[Camera] -- Video --> PE[Processing Engine] end PE -- Position --> Client[Client system] Client -- Control --> PE </pre>
Version 4	Only the camera runs on the mobile device. It communicates remotely with the processing engine running on a separate server, which in turn communicates with a remote client.	 <pre> graph LR subgraph Device Camera[Camera] end subgraph Server PE[Processing Engine] end subgraph Client CS[Client system] end Camera -- Video --> PE PE -- Position --> CS CS -- Control --> PE </pre>

MINIMUM SYSTEM REQUIREMENTS

Server:

- Ubuntu 16.04 , MacOS High Sierra, or above.
- Real quad-core CPU (*AMD Ryzen 3* or *Intel Core i5-8250U*).
- 16 GB of RAM.

Camera(s):

- Monocular camera: 640x480 @ 60fps, field-of-view superior to 70 degrees.
- Stereo camera(s): 2x 640x480 @ 60fps, field-of-view inferior to 120 degrees each.