Visual Behaviour

ASIMROS

Juan Simó, Lara Poves, Óscar Martínez y Rubén Montes

Robótica Software
Universidad Rey Juan Carlos

Mar. 18, 2022

Summary

1. Follow Ball
2. Follow Person
3. Follow Ball&Person

Follow Ball

Behaviour Tree

```
<root main_tree_to_execute="ball_bt">
  <BehaviorTree ID="ball_bt">
      <Sequence name="root_rsequence">
          <ReactiveFallback name="ball_seen">
              <FindBall name="ball_seen"/>
              <Turn name="turn"/>
          </ReactiveFallback>
          <FollowPoint name="follow_point"/>
      </Sequence>
  </BehaviorTree>
</root>
```

Comportamiento

Find Ball

Si puede encontrar la transformada a la bola devuelve SUCCESS

Turn

En caso de no haberla encontrado no sale del Reactive Fallback y gira para reposicionarse

Follow Point

Una vez encuentra la bola la sigue hasta que la pierde, devuelve FAILURE y retoma la búsqueda

Comportamiento

Find Ball

Si puede encontrar la transformada a la bola devuelve SUCCESS

Turn

En caso de no haberla encontrado no sale del Reactive Fallback y gira para reposicionarse

Follow Point

Una vez encuentra la bola la sigue hasta que la pierde, devuelve FAILURE y retoma la búsqueda

Comportamiento

Find Ball

Si puede encontrar la transformada a la bola devuelve SUCCESS

Turn

En caso de no haberla encontrado no sale del Reactive Fallback y gira para reposicionarse

Follow Point

Una vez encuentra la bola la sigue hasta que la pierde, devuelve FAILURE y retoma la búsqueda

Find Ball

Como puede observarse usamos transformadas para encontrar y seguir la pelota

```
BT::NodeStatus
FindBall::tick()
{
  tf2_ros::Buffer buffer;
  tf2_ros::TransformListener listener(buffer);
  if (buffer.canTransform("base_footprint", "ball/0",
          ros::Time(0), ros::Duration(1.0), &error))
  {
    ROS_INFO("He visto la bola");
    return BT::NodeStatus::SUCCESS;
  }
  ROS_INFO("No he visto la bola");
  return BT::NodeStatus::FAILURE;
}
```

Follow Point

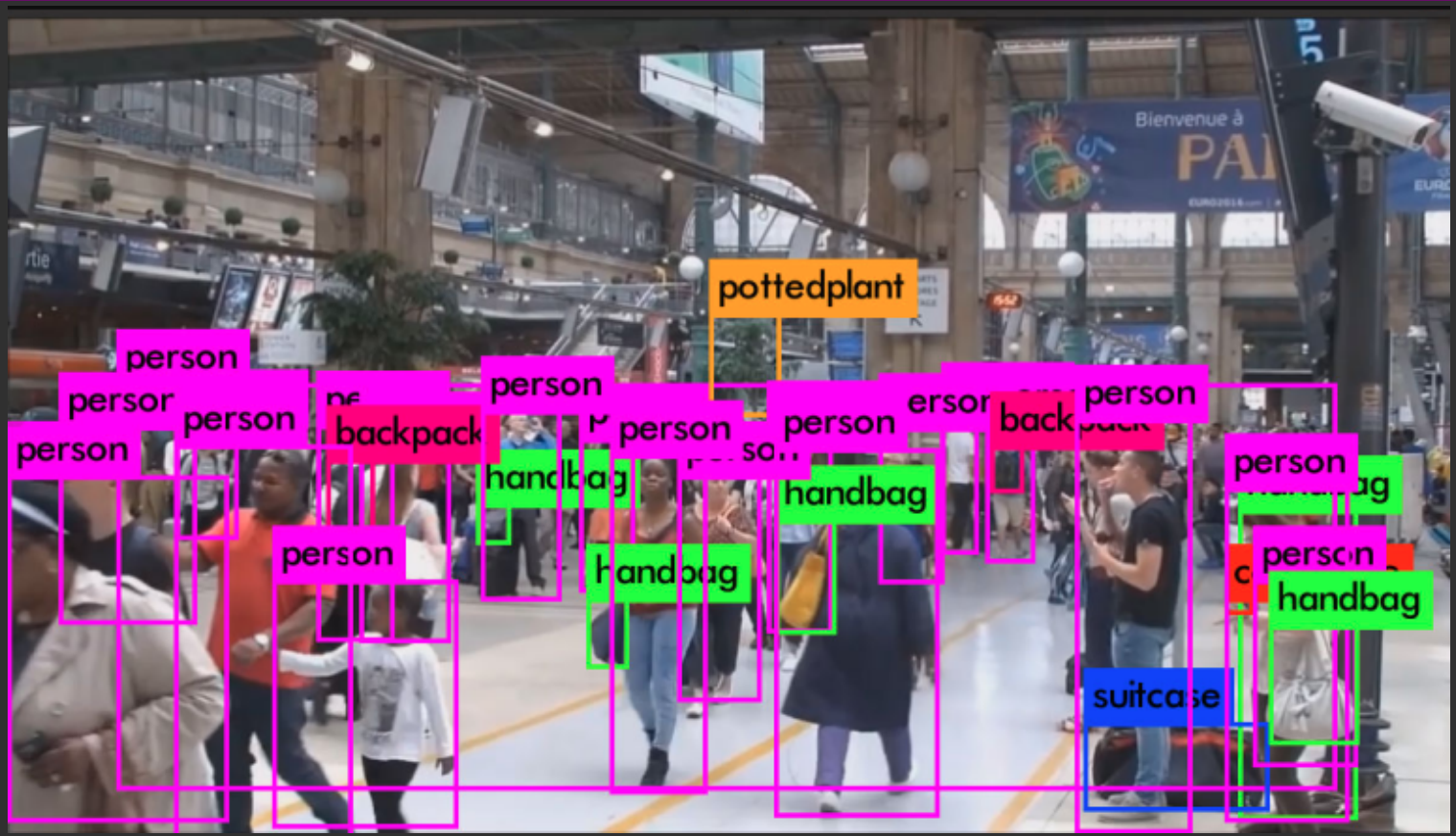
A la hora de seguir la transformada usamos un PID para evitar movimientos bruscos

```
BT::NodeStatus
FollowPoint::tick()
{
  tf2_ros::Buffer buffer;
  tf2_ros::TransformListener listener(buffer);

  if (buffer.canTransform("base_footprint", "ball/0",
        ros::Time(0), ros::Duration(1.0), &error))
  {
    bf2ball_msg = buffer.lookupTransform("base_footprint",
                         "ball/0", ros::Time(0));

    tf2::fromMsg(bf2ball_msg, bf2ball);

    double dist = bf2ball.getOrigin().length();
    double angle = atan2(bf2ball.getOrigin().y(), bf2ball.getOrigin().x());
    vel_msgs_.linear.x = pos_pid_->get_output(dist - 1.0);
    ROS_INFO("%f",  angle_pid_->get_output(angle));
    vel_msgs_.angular.z = angle_pid_->get_output(angle);
    vel_pub_.publish(vel_msgs_);
    return BT::NodeStatus::RUNNING;

  }
  else
  {
    return BT::NodeStatus::FAILURE;
  }
}
```

Follow Person

Follow Person

Para encontrar y seguir a la persona usamos Darknet Ros



Behaviour Tree

```
<root main_tree_to_execute="person_bt">
  <BehaviorTree ID="person_bt">
      <Sequence name="root_rsequence">
          <ReactiveFallback name="person_seen">
              <FindPerson name="find_person"/>
              <Turn name="turn"/>
          </ReactiveFallback>
          <FollowPerson name="follow_person"/>
      </Sequence>
  </BehaviorTree>
</root>
```

Comportamiento

Find Person

Si puede encontrar la transformada a la persona devuelve SUCCESS

Turn

En caso de no haberla encontrado no sale del Reactive Fallback y gira para reposicionarse

Follow Person

Una vez encuentra a la persona la sigue hasta que la pierde, devuelve FAILURE y retoma la búsqueda

Comportamiento

Find Person

Si puede encontrar la transformada a la persona devuelve SUCCESS

Turn

En caso de no haberla encontrado no sale del Reactive Fallback y gira para reposicionarse

Follow Person

Una vez encuentra a la persona la sigue hasta que la pierde, devuelve FAILURE y retoma la búsqueda

Comportamiento

Find Person

Si puede encontrar la transformada a la persona devuelve SUCCESS

Turn

En caso de no haberla encontrado no sale del Reactive Fallback y gira para reposicionarse

Follow Person

Una vez encuentra a la persona la sigue hasta que la pierde, devuelve FAILURE y retoma la búsqueda

Find Person

Gracias a darknet en un Pos_person.h conseguimos la posición a la persona

```
BT::NodeStatus
FindPerson::tick()
{
  dist_ = pos_person.x();
  y_ = pos_person.y();
  if(!std::isnan(dist_) && !(dist_ <= 0.0) &&
    ((ros::Time::now()-pos_person.getTime()).toSec()) < 1.0)
  {
    return BT::NodeStatus::SUCCESS;
  }
  else {
    return BT::NodeStatus::FAILURE;
  }
}
```

Follow Person

Con la posición de la persona y un PID seguimos a la persona hasta perderla y buscar de nuevo

```
BT::NodeStatus
FollowPerson::tick()
{
  dist_ = pos_person.x();
  y_ = pos_person.y();
  if(!std::isnan(dist_) && !(dist_ <= 0.0) &&
    ((ros::Time::now()-pos_person.getTime()).toSec()) < 1.0)
  {
    vel_msgs_.linear.x = pos_pid_->get_output(dist_ - 1.0);
    vel_msgs_.angular.z = angle_pid_->get_output(340 - y_);
    vel_pub_.publish(vel_msgs_);
    return BT::NodeStatus::RUNNING;

  }
  else
  {
    vel_msgs_.linear.x = 0.0;
    vel_msgs_.angular.z = 0.0;
    vel_pub_.publish(vel_msgs_);
    return BT::NodeStatus::FAILURE;
  }
}
```

Follow Ball&Person

Behaviour Tree

```
<root main_tree_to_execute="person_bt">
  <BehaviorTree ID="person_bt">
      <Sequence name="root_rsequence">
          <ReactiveFallback name="find_something">
              <FindBall name="find_ball"/>
              <ReactiveSequence name="find_person_seq">
                  <ReactiveFallback name="find_person_fb">
                    <FindPerson name="find_person"/>
                    <Turn name="turn"/>
                  </ReactiveFallback>
                  <FollowPerson name="follow_person"/>
              </ReactiveSequence>
          </ReactiveFallback>
          <FollowPoint name="follow_balll"/>
      </Sequence>
  </BehaviorTree>
</root>
```

Comportamiento

- Primero busca la pelota
- En caso de no encontrarla busca a la persona
- Si tampoco la encuentra gira y vuelve a empezar
- En caso de encontrar la pelota nunca enra a la persona y sigue la pelota
- Si enuentra a la persona entonces sigue a esta

The End