

---

**Team 12**

---

Frédéric Blondeel	h
Martijn Debeuf	h
Toon Sauvillers	h
Dirk Vanbeveren	h
Bert Van den Bosch	h
Seppe Van Steenberghe	h

---

---

## Inhoudsopgave

<b>1</b>	<b>Reconstructie</b>	<b>2</b>
1.1	test . . . . .	2

# 1 Reconstructie

Na het filteren van de hoeken kan het voorkomen dat er geen vier meer overblijven. Dit wil zeggen dat er mogelijks overlap van schermen in de opstelling voorkomt of dat een of meerdere hoeken gewoon niet zichtbaar is door een obstakel. Er wordt verondersteld dat minstens twee hoekpunten en het middelpunt zichtbaar zijn. Indien men zich aan deze eis houdt kan met het hieronder uitgelegde algoritme steeds mogelijk de andere hoeken te reconstrueren. Eerst Volgt een korte opsomming van de stappen daarna worden deze meer in detail besproken. De input die meegegeven is een dictionary. De sleutels zijn LU,RU,RD en LD m.a.w. posities van hoeken en als bijhorende waarden coördinaten voor deze die reeds gevonden zijn en een *null* als plaatshouder voor de nog te reconstrueren hoeken. Eerst wordt vanuit het middelpunt de vier punten bepaald die op de diagonalen liggen. Deze worden ook in dictionary opgeslaan met dezelfde structuur als de input. Daarna wordt hoek per hoek gekeken welke nog ontbreken. Indien reconstructie nodig is, wordt het overeenkomende punt van de diagonalen genomen. Samen met het middelpunt wordt hieruit een eerste reconstructielijn opgesteld. Door de vooropgestelde eis zal er altijd een van de aanliggende hoeken wel gedetecteerd zijn. Nu wordt ook vanuit dit punt de uiterste pixels van de zijden bepaald. Na bepaling van het correcte uiteinde kan de

## 1.1 test

## Referenties