

Bild 5.49 Erzeugung der visuellen Hülle aus zwei Kameraperspektiven in vereinfachter 2D-Darstellung: Zwei Silhouettenbilder S_1 und S_2 markieren Sichtstrahlen mit Hindernissen (menschliche Arbeiter, orange) als Vordergrund (flache Leisten, weiß), alle anderen Sichtstrahlen als Hintergrund H (flache Leisten, schwarz). Anfangs sind alle Voxel als unbekannt (grau) markiert (oben links). Jeder Voxel ist mit allen Sichtstrahlen abzugleichen (oben rechts): Voxel, die in zumindest einem Sichtstrahl als Hintergrund markiert sind, enthalten kein Hindernis und bleiben im Ergebnis hindernisfrei (schwarz). Liegt ein Voxel dagegen nur auf Vordergrundsichtstrahlen (unten links), ist der Voxel als Hindernis anzunehmen (weiß), unabhängig davon, ob der Voxel tatsächlich ein Hindernis enthält. Somit entstehen Pseudohindernisse (blaue Wolken), die die Überwachungslösung im Ergebnis (unten rechts) nicht als Freiraum bestätigen kann.

den. Für jede Kamera seien zusätzlich die extrinsischen wie intrinsischen Kameraparameter bekannt. Durch die Kameraparameter ist es leicht möglich, zwischen dem zu prüfenden Voxel und den Sichtstrahlen aller Kameras abzugleichen. Zum Abgleich ist schließlich ein sicherheitsorientiertes Vorgehen nötig, um die vollständige visuelle Hülle zu ermitteln: Ein Voxel zählt genau dann als frei, wenn keine Kamera auf Sichtstrahlen durch den Voxel ein Hindernis beobachtet, wenn also alle Sichtstrahlen durch den Voxel zu Hintergrundpixeln gehören. Ein Voxel ist dagegen genau dann als potenziell hindernisbelegt zu markieren, wenn irgendein hindernisbehafteter Sichtstrahl irgendeiner Kamera den Voxel schneidet, wenn also irgendein Sichtstrahl durch den Voxel zu einem Vordergrundpixel gehört. Etwas formaler ist dieses Vorgehen in Listing 5.8 aufbereitet, ein passendes Beispiel ist in Bild 5.49 dargestellt.

Listing 5.8 Pseudocode für die visuelle Hülle

```

01 Eingabe: Silhouettenbilder aus Multikamerasystem
02   Voxel des Arbeitsraums
03 Ausgabe: Hindernis-Zustand für jeden Voxel
04 FOR EACH voxel v IN arbeitsraum DO
05 | v.hindernis = unbekannt
06 FOR EACH voxel v IN arbeitsraum DO
07 | ; Setze Voxel auf freiem Sichtstrahl als frei.
08 | FOR EACH kamera k IN multikamerasystem DO
09 | | FOR EACH pixel p IN k DO
10 | | | IF v IN p.sichtstrahl AND p = hintergrund THEN
11 | | | | v.hindernis = false
12 |
13 | ; Voxel auf keinem freien Sichtstrahl ist Hindernis.
14 | IF v.hindernis = unbekannt THEN
15 | | v.hindernis = true

```

Die Verwendung mehrerer Kameras ermöglicht es dem Fusionsverfahren, den Arbeitsraum des Roboters in vollem *Umfang* zu überwachen. Zudem erlaubt der Abgleich der Kameras, die Verdeckungen durch ein Hindernis in einer Kamera über den Blickwinkel einer

anderen Kamera zu verringern. Das steigert die *Robustheit* der Lokalisation. Allerdings bleibt die Hindernislokalisation immer noch auf die visuelle Hülle der Hindernisse begrenzt. Insbesondere können Verdeckungen zwischen zwei getrennten Hindernissen trotzdem auftreten. Dadurch verbleiben *Pseudohindernisse*, also Bereiche innerhalb von Verdeckungen bezüglich aller Kameras. Diese Pseudohindernisse können den Roboter in seiner Reaktionsplanung deutlich einschränken. Ein Beispiel für Pseudohindernisse findet sich in Bild 5.49.

Neben erhöhtem Überwachungsumfang und erhöhter Robustheit bietet eine Lösung mit mehreren Kameras gegenüber einer Überwachung mit einzelner Kamera erstmals eine Lokalisation mit *Redundanz*: Fällt eine der Kameras in einem Multikamerasystem aus, wird der Roboter zwar in seiner Reaktion eingeschränkt, jedoch bleibt der Roboter im Rahmen der eingeschränkten Überwachungslösung weiterhin einsatzbereit.

Einerseits bietet die Hindernislokalisation über die Sensorfusion mehrerer Kameras zwar deutliche Vorteile gegenüber Varianten mit nur einer Kamera, andererseits sind jedoch auch einige Nachteile festzustellen: Mit zunehmender Zahl an Kameras steigen die *Kosten* der Überwachungslösung. Zudem bedingt die Echtzeitverarbeitung vieler Kamerabilder eine hohe *Rechenzeit*, die auf einzelnen Rechnern nicht mehr sinnvoll zu bewältigen ist. Die damit einhergehende Verteilung der Last auf ein verteiltes Rechnersystem erhöht nicht nur weiter die Kosten der Lösung, sondern steigert den *Implementierungsaufwand*. Selbst bei Verwendung weniger Kameras sind die zugrundeliegenden Algorithmen, inklusive der nötigen Programmoptimierung, deutlich komplexer als bei der Überwachung mittels Einzelkamera, ein weiterer Beitrag zum hohen Implementierungsaufwand. Ferner fordert die Komplexität der Mehrkameralokalisation beim Einrichten einen Mehraufwand ein: Der *Einrichtaufwand* steigt einerseits durch das (teilweise automatisch mögliche) Platzieren der Kameras in der Arbeitszelle, denn es müssen für alle Kameras sinnvolle und möglichst verdeckungsfreie Sichtbereiche gefunden werden. Andererseits ist zur intrinsischen wie extrinsischen Kamerakalibrierung ein aufwendigeres Vorgehen nötig, da im Allgemeinfall die unvollständige Überdeckung der Kamerasichtbereiche eine gleichzeitige Kalibrierung aller Kameras verhindert. Am erhöhten Einrichtaufwand lässt sich leicht eine erhöhte *Pervasiveness* ausmachen, denn gegebenen-

falls ist für eine verdeckungsfreie Platzierung aller Kameras die Einrichtung der Arbeitszelle anzupassen. Zuletzt ist ein hoher *Betriebsaufwand* zu befürchten: Das Rechnersystem ist zu warten, durch die steigende Zahl an Geräten sind zu behebende Ausfälle einzelner Geräte wahrscheinlicher, und bei Änderungen an der Arbeitszelle fällt für die Aufnahme von Referenzbildern ein höherer Zeitbedarf an.

In der Literatur findet die grundlegende visuelle Hülle zur Hindernislokalisation im Rahmen der Grobbahnplanung für Roboter beispielsweise in (Ladikos et al. 2008) sowie in (Stengel et al. 2012) ihren Einsatz. Eine Variante zur visuellen Hülle unter Präsenz bekannter Verdeckungen ist in (Kuhn, Henrich, 2009) diskutiert.

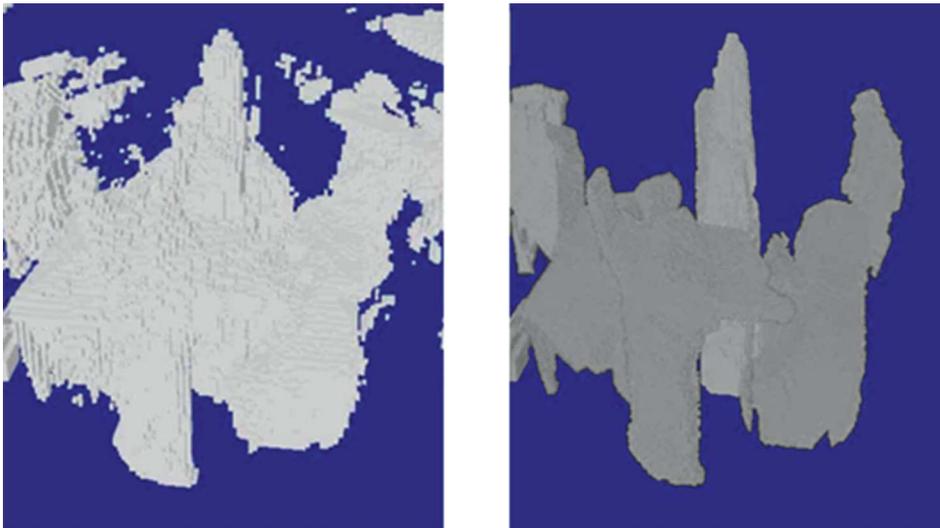
Verfahrensverbesserungen zur visuellen Hülle

Zu Überwachungslösungen mit einem Mehrkamerasystem auf Grundlage der visuellen Hülle gibt es zahlreiche Verfahrensverbesserungen, die bestehende Nachteile ausgleichen oder die bereits gute Qualität der Ausgabe noch verbessern. Im Folgenden sind mit der Nutzung von Tiefenkameras, der Nutzung von Zusatzwissen und der Nutzung alternativer Datenstrukturen drei besonders wichtige Vertreter von Verfahrensverbesserungen kurz skizziert und bewertet.

Nutzung von Tiefenkameras für die visuelle Hülle

Zur Nutzung von Tiefenkameras anstelle von oder zusätzlich zu Farbkameras ist lediglich eine einfache Erweiterung des bestehenden Verfahrens für die visuelle Hülle nötig. Bei dieser Erweiterung werden Voxel auch dann als Freiraum markiert, wenn sie aus Sicht einer beliebigen Kamera auf allen Sichtstrahlen dieser Kamera vor den zugehörigen Tiefenwerten liegen. Dieses Vorgehen ist zulässig, da die Kamera vollständig durch den Voxel hindurchblicken konnte, denn die Kamera hat Tiefenwerte nur für ein weiter hinten liegendes Hindernis bestimmt. Bei dem Voxel muss es sich folglich um Freiraum handeln.

Die Verwendung von Tiefenkameras als *Eingabe* der Sensorfusion zur visuellen Hülle erlaubt durch die zusätzliche Markierung von Voxeln als Freiraum und durch die dann unnötige Vordergrund-Hintergrund-Trennung nochmals einen deutlichen Zugewinn an Robustheit. Dem gegenüber steht eine durch die datenreicheren Punktwolken gering erhöhte *Rechenzeit* und ein leicht erhöhter *Implementierungsaufwand*. In den *anderen Kriterien* ist durch die Verwendung von Tiefenkameras auch bei der Sensorfusion zur visuellen Hülle

**Bild 5.50**

Vergleich der gewöhnlichen visuellen Hülle (links) mit wissensbasierter Verfeinerung (rechts): Die durch Verdeckungen verursachten Pseudohindernisse sind in der rechten Variante deutlich reduziert. Die Silhouetten der Menschen (links, rechts) und des Roboters (mittig) sind jetzt deutlich zu erkennen.

keine prägnante Änderung zu verzeichnen. Abschließend sei jedoch der Hinweis gegeben, dass mehrere Tiefenkameras (je nach Messprinzip) für Interferenzen anfällig sein können. Das kann eine kombinierte Lösung aus einer Tiefenkamera (z. B. für den Hauptaufenthaltsbereich des Menschen) und Farbkameras (z. B. für Randbereiche der Arbeitszelle) sinnvoll machen. Wissenschaftliche Arbeiten zur visuellen Hülle aus Tiefenkameras sind (Fischer, Henrich, 2009; Alexiadis et al. 2013), und (Lenz 2012).

Wissensbasierte Verfeinerung der visuellen Hülle

Neben der Verwendung von Tiefenkameras kann die visuelle Hülle durch die Nutzung wissensbasierter Ansätze in ihrer Robustheit verbessert werden. Grundlage der wissensbasierten Ansätze bildet die Beobachtung, dass die visuelle Hülle trotz der allgemein schon hohen Robustheit für bestimmte Arten von Verdeckungen anfällig bleibt: Entsteht auf einer der Kameras (z. B. durch Kameraräuschen oder durch Verschmutzung der Linse) ein kleiner, aber als falsch-positiv eingeordneter Vordergrundbereich, kann dieser zusammen mit anderen korrekt oder falsch-positiven Vordergrundbereichen zu kleinen Pseudohindernissen im Arbeitsraum des Roboters führen. Ein menschlicher Beobachter ist, im Gegensatz zum Computerprogramm, jedoch in der Lage die kleinen Pseudohindernisse als solche zu erkennen. Dazu nutzt der Mensch bereits vorhandenes Zusatzwissen: Der Mensch weiß, dass Hindernisse nicht in der Luft fliegen, dass Hindernisse eine gewisses Minimalvolumen oder eine gewisse Minimalausdehnung besitzen, oder dass Hindernisse nicht aus

dem Nichts auftauchen können. Bei wissensbasierten Ansätzen ist nun die Idee, einen Teil des menschlichen Zusatzwissens in den Computer einzuprogrammieren. Dazu gehört beispielsweise, dass fliegende Hindernisse zu verwerfen sind, oder dass Hindernisse sich nur vom Rand der Arbeitszelle aus in den Kamerabereich begeben können. Das einprogrammierte Wissen ermöglicht dem Rechner schließlich als nachgeschalteten Schritt, die mit herkömmlichen Verfahren ermittelte visuelle Hülle bezüglich der Pseudohindernisse zu optimieren. Ein Vergleich zwischen gewöhnlichem Vorgehen und visueller Hülle mit wissensbasierter Verfeinerung findet sich in Bild 5.50.

Insgesamt ist durch wissensbasierte Ansätze also eine Erhöhung der *Robustheit* zu erreichen. Auf Seite der Nachteile bleiben der hohe *Implementierungsaufwand* sowie der je nach Zusatzwissen extrem hohe Bedarf an *Rechenzeit* zu nennen. Auf *andere Kriterien* der visuellen Hülle hat eine nachgeschaltete, wissensbasierte Verfeinerung keinen Einfluss.

Zuletzt sei als Spezialfall der wissensbasierten Verfeinerung die Verwendung von CAD-Modellen zur Filterung im Voraus bekannter Objekte erwähnt. Bislang ist bei der Vordergrund-Hintergrund-Trennung ein kompliziertes Vorgehen (z. B. mittels Wörterbuch-Referenzbildern) nötig, um Sichtstrahlen auf solche Objekte, auf festinstallierte Teile der Arbeitszelle oder auf den Roboter selbst zu filtern, etwa damit der Roboter durch eine fälschliche Selbsteinstufung als Hindernis nicht ständig blockiert. Bei der Lokalisation mit Sensorfusion eröffnen sich durch die Verfügbarkeit einer 3D-Darstellung der Hindernisse zusätzliche Alternati-

ven, um im Voraus bekannte Objekte effizient aus der Ausgabe zu filtern. Insbesondere kann als zusätzliches Wissen für jedes bekannte Objekt ein passendes CAD-Modell inklusive zugehöriger Pose Eingang in den wissensbasierten Ansatz finden. Als Hindernis markierte Voxel, die in einem CAD-Modell liegen, können bei dieser Variante stattdessen als Freiraum markiert werden. Der Roboter kann dann wahlweise entweder das besonders genaue CAD-Modell (z. B. für zu greifende Objekte) oder Freiraum (z. B. für den Roboter in aktueller Pose) als Grundlage für die Reaktionsplanung heranziehen. Dieses Vorgehen führt zu einer weiteren Steigerung der *Robustheit* auf Kosten der *Rechenzeit*. Das Erfassen von CAD-Modellen der festmontierten Arbeitszelle kann, sofern die Modelle nicht ohnehin vorliegen, einen zusätzlichen *Einrichtaufwand* und *Betriebsaufwand* nach sich ziehen. *Weitere Kriterien* bleiben von der wissensbasierten Verfeinerung mittels CAD-Modellen unberührt.

Die Erstellung einer wissensbasierten Verfeinerung für eine gegebene visuelle Hülle ist Gegenstand der Arbeit (Kuhn 2012).

Alternative Datenstrukturen und Algorithmen für die visuelle Hülle

Als letzte Verfahrensverbesserung zur visuellen Hülle sei die Nutzung geeigneter Implementierungsvarianten angesprochen. Motivation zur Diskussion der Implementierung liefert die Erkenntnis, dass jede Steigerung der *Robustheit* mit einer erhöhten *Rechenzeit* einhergeht. Diesem Zusammenhang ist nun auf Implementierungsebene entgegenzuwirken: Entweder, um Kosten für Hardware durch geringere Rechenzeit bei gleicher Robustheit zu sparen, oder um bei gegebener Rechenzeit und gegebenen Kosten eine höhere Robustheit zu erreichen.

Um das Ziel reduzierter *Rechenzeit* bei gleicher Ausgabe zu erreichen, ist zwingend eine Anpassung an verwendete Algorithmen und Datenstrukturen nötig. Grundlage dafür bilden zwei Beobachtungen: Zum einen nehmen Hindernisse und Freiraum sowohl in Eingabe wie in Ausgabe große und zusammenhängende Raumbereiche ein, zum anderen verändern sich Eingabe wie Ausgabe über kurze Zeit hinweg nur wenig. Die erste Beobachtung ist anschaulich schnell klar: Menschen belegen, ebenso wie eingebrachte Objekte, immer einen größeren Bereich, dazwischen gibt es ebenfalls große leere Raumbereiche. Der Gegenfall, dass sehr viele, sehr kleine Objekte wie fliegendes Konfetti in der Arbeits-

zelle verteilt sind, ist dagegen in den allermeisten Anwendungen auszuschließen. Die zweite Beobachtung bedarf einer Observation der Arbeitszelle über mehrere Kamerabilder hinweg: Bewegt sich ein Hindernis, so kommen in der Eingabe von Kamerabild zu Kamerabild nur wenige neue Vordergrundpixel hinzu, und es fallen nur wenige Hintergrundpixel weg. Analog dazu sind in der Ausgabe nur wenige Voxel neu als Hindernis oder als Freiraum zu markieren. Bei einer vollkommen reigungslosen Umgebung des Roboters ist, als Extremfall, nach der ersten Rekonstruktion überhaupt kein Zusatzaufwand nötig: Weder Eingabe noch Ausgabe ändern sich. In beiden Fällen spricht die Informatik von einer *Kohärenz* – einem örtlichen oder zeitlichen Zusammenhang, der zum Entwurf schnellerer Algorithmen und Datenstrukturen genutzt werden kann. Aus der örtlichen und aus der zeitlichen Kohärenz erwächst jeweils eine eigene Variante der visuellen Hülle.

Zur Ausnutzung der örtlichen Kohärenz ist es nötig, Eingabe wie Ausgabe in größere Bereiche zu zerlegen. Auf Seite der *Eingabe* fällt es leicht, zusammenhängende Bereiche schrittweise zusammenzufassen: Aus einem quadratischen Block aus vier Pixeln entsteht ein zusammenhängender Bereich, wenn die Pixel im Bereich entweder alle als Vordergrund oder alle als Hintergrund markiert sind. Dieses Vorgehen setzt sich iterativ auf den zusammengefassten Bereichen fort: Ist ein quadratisches Stück aus vier solcher Bereiche ebenfalls entweder in allen Bereichen als Vordergrund oder in allen Bereichen als Hintergrund markiert, werden diese Bereiche erneut zu einem noch größeren Bereich zusammengefasst. Das Vorgehen endet erst dann, wenn kein weiteres Zusammenfassen mehr möglich ist. Das Zusammenfassen der Eingabe an einer Beispielsilhouette findet sich in Bild 5.51 dargestellt.

Das Zusammenfassen von Bereichen der Eingabe geschieht in einem separaten Vorverarbeitungsschritt. Die zusammengefassten Bereiche stehen anschließend als zusätzliche Eingabe zum Erstellen der visuellen Hülle zur Verfügung. Insbesondere ist für jeden zusammengefassten Bereich anstelle der vielen, einzelnen Sichtstrahlen durch Kamerapixel im Bereich nur noch ein einzelner, großer Kamerasichtstrahl durch alle Pixel des Bereichs zu berücksichtigen.

Auf Seite der *Ausgabe* ist ein geschicktes Zusammenfassen nicht möglich, da die Ausgabe dazu erst einmal vollständig bestimmt werden müsste. Stattdessen ist ein umgekehrtes Verfahren nötig: Man beginnt mit einem großen, zusammenhängenden Bereich, der den gesamt-



Bild 5.51 Iteratives Zusammenfassen von Eingabepixeln (links) einer Silhouette zu größeren Raumbereichen (weitere Bilder), die eindeutig Vordergrund (weiß) oder Hintergrund (schwarz) sind. Falls kein weiteres Zusammenfassen möglich ist, müssen in dem entsprechenden Bereich (grau) die vorherigen Pixel oder Bereiche beibehalten werden.

ten, überwachten Bereich abdeckt. Dieser Bereich wird nun entsprechend dem Verfahren der visuellen Hülle gegen die durch Vorverarbeitung zusammengefassten Sichtstrahlen geprüft: Sollten alle – gegebenenfalls zusammengefassten – Sichtstrahlen in diesem Raumbereich durch Vordergrundpixel gehen, ist der Bereich komplett gefüllt. Gehen dagegen alle Sichtstrahlen einer Kamera im Raumbereich durch Hintergrundpixel, ist der Raumbereich vollständig leer. In allen anderen Fällen ist keine eindeutige Aussage möglich, es finden sich in dem Raumbereich sowohl Hindernisse als auch Frei-

raum. Der Raumbereich ist also in kleinere Raumbereiche aufzuteilen, die erneut zu prüfen sind. Das Aufteilen setzt sich ebenfalls iterativ fort, bis weiteres Aufteilen keinen zusätzlichen Informationsgewinn mehr bietet. Alle dann noch verbleibenden Raumbereiche müssen aus Sicherheitsgründen als Hindernis betrachtet werden. Bild 5.52 illustriert das Aufspalten der Ausgabe in immer feinere Bereiche.

Das Ausnutzen der örtlichen Kohärenz ist zuletzt leicht mit der Ausnutzung zeitlicher Kohärenz zu kombinieren. Dazu ist für jeden Eingabebereich festzuhalten, ob

5

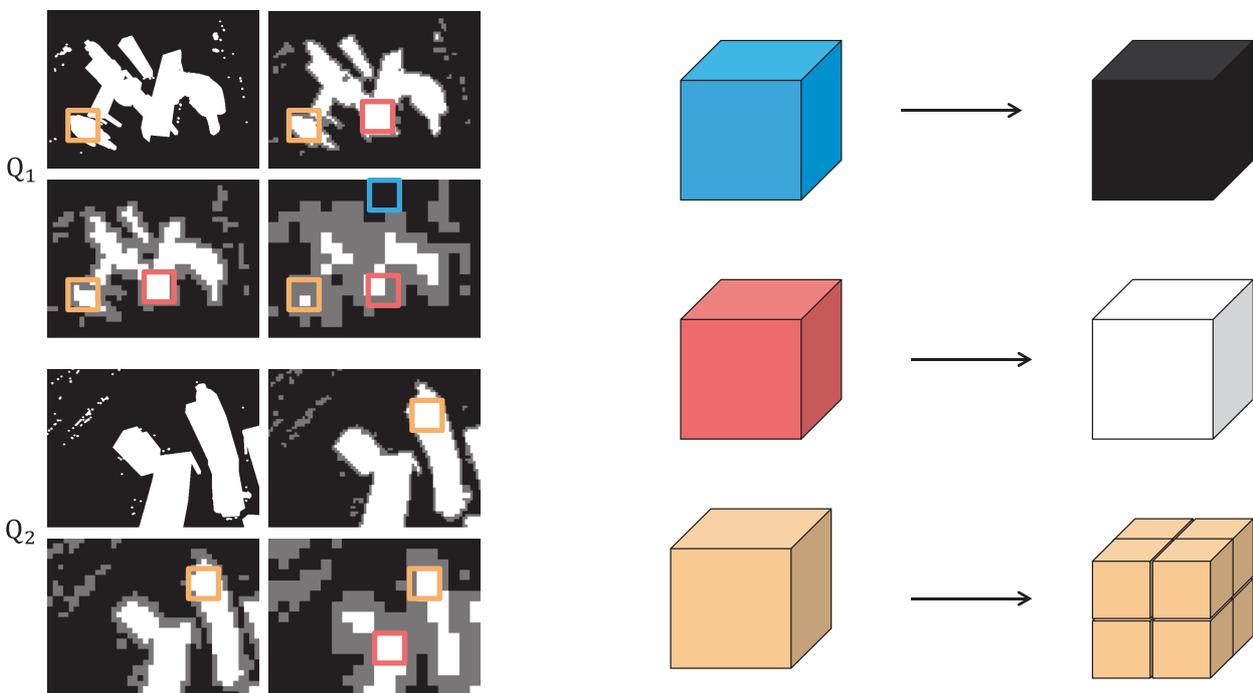


Bild 5.52 Vorgehen beim Aufspalten der Ausgabe in einzelne Bereiche (rechts) auf Grundlage bereits zusammengefasster Eingabebereiche (links) in Kamerabildern Q_1 und Q_2 : Ein gegebener Ausgabebereich (blauer Würfel) ist vollständig hindernisfrei, sofern alle zugehörigen Sichtstrahlen einer Kamera als Hintergrund markiert sind. Dazu genügt es, entsprechende zusammenhängende Eingabebereiche zu prüfen (blaues Quadrat). Dagegen ist ein Ausgabebereich (roter Würfel) vollständig als Hindernis zu betrachten, sofern zugehörige Sichtstrahlen in zusammenhängenden Bereichen aller Kameras Vordergrund sind (rote Quadrate). In allen anderen Fällen (gelbe Quadrate) ist der Ausgabebereich (gelber Würfel) in Teilbereiche zu zerlegen, die nach demselben Muster erneut zu prüfen sind.

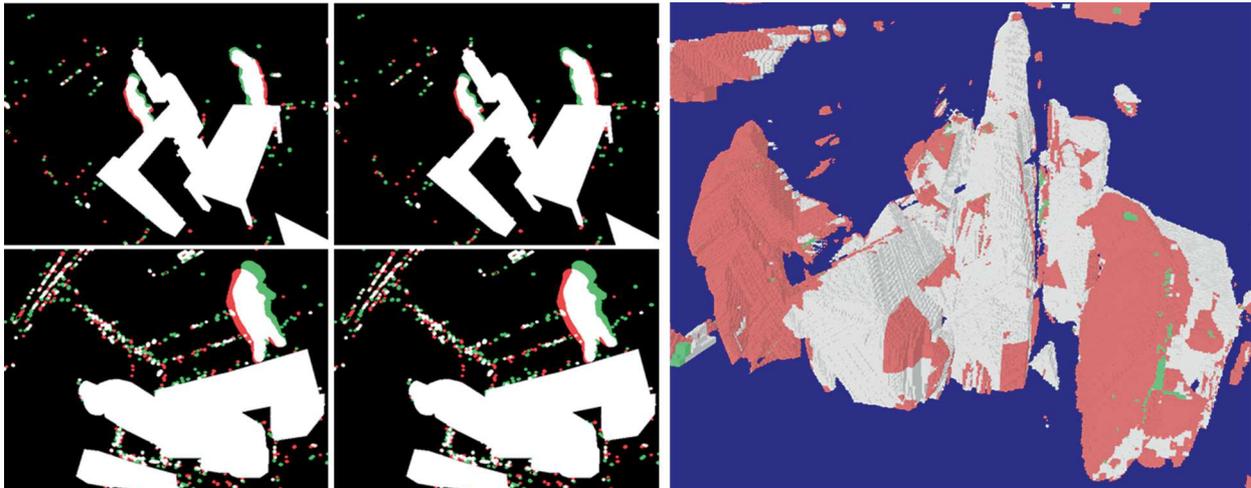


Bild 5.53 Bestimmung einer visuellen Hülle (rechts) aus zusammenhängenden Eingabebereichen eines Mehrkamerasystems mit vier Kameras (links, Silhouetten). In der Eingabe sind geänderte Bereiche markiert, die erst im aktuellen Bildersatz Vordergrund (grün) oder Hintergrund (rot) sind. Alle schwarzen Bereiche (Hintergrund) und weißen Bereiche (Vordergrund) sind seit dem letzten Bildersatz des Multikamerasystems unverändert. In der Ausgabe setzt sich die Markierung fort: Große Bereiche der visuellen Hülle (weiß) sind unverändert übernommen, nur Bereiche, die jetzt ein Hindernis enthalten (grün), oder Bereiche, die jetzt Freiraum sind (rot), mussten angepasst werden.

sich dieser Bereich im Vergleich zum vorangehenden Kamerabild verändert hat. Auf Grundlage der unveränderten Eingabebereiche ist es schließlich möglich, zusammenhängende Bereiche einer vorherigen Ausgabe beizubehalten. Ein zusammenhängender Bereich der Ausgabe ist insbesondere dann beizubehalten, wenn sich kein über Sichtstrahlen korrespondierender Bereich der Eingabe geändert hat. Durch die Verwendung und Weiterverwendung von Bereichen sinkt die Zahl an Berechnungen, die sonst zum immer neuen Bestimmen einer visuellen Hülle aufgewendet werden müsste. Das entsprechende Vorgehen ist in Bild 5.53 skizziert.

Eine Bewertung der verbesserten Algorithmen und Datenstrukturen bestätigt die erhofften Vorteile: Durch die Ausnutzung von räumlicher und zeitlicher Kohärenz sinkt die benötigte *Rechenzeit* deutlich. Ebenso deutlich steigt jedoch mit dem jetzt komplexeren Vorgehen der *Implementierungsaufwand*. Auf *weitere Bewertungskriterien* hat das neue Vorgehen gegenüber der naiven visuellen Hülle keinen Einfluss. Zuletzt ist als kleiner Ausblick festzuhalten, dass die zusammengefassten Ausgabebereiche im Rahmen der späteren Roboterreaktion deutlich einfacher auszuwerten sind und damit dort ebenfalls Rechenzeit einsparen.

Beispiele für alternative Datenstrukturen und Algorithmen finden sich in (Bigdelou et al. 2009; Soares 2007) und (Werner, Henrich 2014).

Sensorfusion zur Photohülle

Neben der visuellen Hülle gibt es verschiedene Alternativverfahren, die ebenfalls eine besonders robuste Lokalisation von Hindernissen in der Arbeitszelle versprechen. Im Folgenden werden zwei solcher Alternativen betrachtet: die Sensorfusion mittels Photohülle und die Sensorfusion zur probabilistischen Lokalisation.

Die Sensorfusion mittels Photohülle ähnelt dem Vorgehen bei der Lokalisation mit visueller Hülle: Erneut wird der Arbeitsraum des Roboters in Voxel zerlegt, für die das Verfahren der Photohülle als *Ausgabe* jeweils entweder Hindernisfreiheit oder eine mögliche Hindernisbelegung bestimmt. Im Gegensatz zur visuellen Hülle unterstützt die Photohülle allerdings direkt die Nutzung der Farbinformationen in eingehenden Kamerabildern als *Eingabe*. Dadurch entfällt die aufwendige und bisweilen fehleranfällige Vorverarbeitung durch Vordergrund-Hintergrund-Trennung, Referenzbilder werden ebenfalls unnötig.

Zur direkten Fusion der Farbinformation über die Farbbilder verschiedener Kameras definiert die Photohülle den Begriff der *Farbkonsistenz*: Ein Voxel ist *farbkonsistent* genau dann, wenn alle Sichtstrahlen durch diesen Voxel ungeachtet der zugehörigen Kameras dieselben Farbwerte aufweisen. Voxel, die diese Eigenschaft nicht erfüllen – also in zumindest einem Kamerabild eine andere Farbe besitzen als in anderen Kamerabildern –, nennt man *farbinkonsistent*. Unter der Annahme, dass Voxel hinreichend klein sind, muss jeder