

Effiziente, parallelisierte Berechnung von Bildkorrespondenzen für automotives Sehen

Michael Schweitzer, Hans-Joachim Wuensche
Institut für Technik Autonomer Systeme (LRT8/TAS)
Universität der Bundeswehr München
85579 Neubiberg
{misw,jw}@unibw.de

Kurzfassung

Dieser Artikel beschreibt ein echtzeitfähiges Verfahren zur Berechnung von Bildkorrespondenzen eines Bildstroms. Dabei liegt der Fokus auf der Parallelisierung durch GPUs und die Spezialisierung auf den automotiven Bildstrom. Es gelingt damit, die Eigenbewegung zu schätzen und Hypothesen für bewegte Objekte zu erzeugen.

1. Einleitung

Bildkorrespondenzen sind die Grundlage für eine Vielzahl von Methoden maschinellen Sehens. Für den automotiven Einsatz gibt es die Anforderung der schnellen und effizienten Berechnung, aber auch wesentliche Merkmale des Bildstroms, welche dieser Anforderung entgegenkommen. In diesem Beitrag wird eine Arbeit [11] vorgestellt, welche eine spezielle Lösung der Berechnung von Bildkorrespondenzen für automotives Sehen darlegt. Aus einem monokularen Bildstrom (Auflösung 768×480) werden Langzeitkorrespondenzen in einer Rechenzeit unter $3ms$ bestimmt. Bei der Auswertung eines Stereo-Bildstroms (Auflösung $2 \times 768 \times 480$) werden Korrespondenzen der beiden monokularen Ströme über der Zeit, sowie des aktuellen Stereo-Bildes ermittelt. Die benötigte Verarbeitungszeit liegt bei unter $6ms$.

Erreicht wird die schnelle Verarbeitung im wesentlichen durch drei Optimierungen: Effiziente Extraktion von Ecken als Schlüsselpunkte basierend auf Integralbildern, effizientes Matching und massive Parallelisierung. Bei der Ermittlung der Schlüsselpunkte wird ausgenutzt, dass im automotiven Bildstrom keine Rotationsinvarianz nötig ist. Das Matching kann basierend auf einer Vorsortierung zu 10 verschiedenen Klassen erheblich beschleunigt werden.

Die Parallelisierung wird mit einer GPU erreicht. Je nach Hardware kann mit einer Anzahl von bis zu 768 Rechenkernen gearbeitet werden. Nicht jeder Algorithmus kann dabei volle

Parallelität erreichen. Das hier vorgestellte Verfahren ist für den Einsatz auf parallelverarbeitenden Architekturen entwickelt und erreicht einen sehr hohen Grad an Parallelität. Demnach wäre auch eine FPGA-Version denkbar.

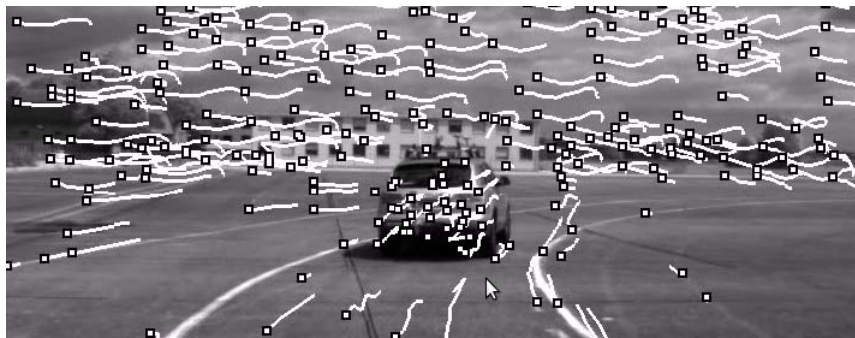


Abbildung 1. Extraktion von Langzeit-Bildkorrespondenzen basierend auf dem *SidCell*-Bild.

Grundlegend für Effizienz des Verfahrens ist das sogenannte *SidCell*¹-Bild, eine Datenstruktur, welche gleichzeitig Extraktor *und* Deskriptor für Bildschlüsselpunkte ist. Zudem erlaubt es eine Vorauswahl von Korrespondenz-Kandidaten, wodurch das Matching erheblich beschleunigt wird. Abbildung 1 zeigt durch das *SidCell*-Bild bestimmte Langzeitkorrespondenzen bei einer Kurvenfahrt nach rechts. Zusätzlich zu [11] zeigen wir, wie durch Hinzunahme von Eigenbewegungsdaten die Robustheit des Verfahrens gesteigert wird (siehe hierzu [10]). Weiterhin verdeutlichen wir die vielseitige Verwendbarkeit des Verfahrens, speziell für automotiv Sichtsysteme. Wir erläutern den Einsatz zur Schätzung der Eigenbewegung und Hypothesen-Generierung von Fahrzeugen. Videos sind verfügbar unter [13].

2. Parallelisierte Berechnung von *SidCell* Bildkorrespondenzen

Die Bestimmung von Bildkorrespondenzen (im Folgenden *BK*) zwischen zwei verschiedenen Kamerabildern ist eine fundamentale Aufgabe in der Bildverarbeitung. Der Anwendungsbereich erstreckt sich von der Kamerakalibrierung über Objekttracking hinzu visuellen Odometrie-Verfahren und hochauflösender 3D-Rekonstruktion. Beispiele für automotiv Anwendungen wären die Online-Kalibrierung einer Stereo-Plattform [4] oder die Erkennung und Posenschätzung entgegenkommender Fahrzeuge mittels Stereo [1]. Je weniger Rechenzeit dabei der *BK*-Prozess benötigt, desto mehr Verarbeitungszeit steht den nachfolgenden Verarbeitungsschritten zur Verfügung. Eine zugegebenermaßen willkürliche, aber intuitive Anforderung an den *BK*-Prozess ist, nicht mehr als 10 Prozent der zur Verfügung stehenden Rechenzeit zu verbrauchen, damit der Großteil den komplexen Aufgaben der Gesamtperzeption und Entscheidungsfindung zukommt. Bei einem Kameratakt von 20Hz stünden also nur 5ms Rechenzeit zur Verfügung.

Im Folgenden wird ein Verfahren vorgestellt, mit welchem basierend auf einer zentralen Bildstruktur (*SidCell-Bild*) entsprechende Laufzeiten erreicht werden.

¹*Scale Invariant Descriptive Cells*

2.1. SidCell-Bild

Der BK-Prozess unterteilt sich in zwei Hauptteile: erstens die Extraktion von Bildschlüssel-
punkten (*Keypoint Extraction*) und die Assoziierung mit Schlüsselpunkten des vorange-
gangenen Bildes (*Keypoint Matching*). Wesentlich für die Effizienz des hier vorgestellten
Verfahrens ist, dass *beide* Schritte auf *einer* Bildstruktur aufbauen. Dies ist ein wesentlicher
Unterschied zu herkömmlichen Verfahren und reduziert den Rechenaufwand.

Viola *et al.* [12] stellt in einer Methode zur Gesichtserkennung in Bildern das Konzept
des *Integralbildes* vor. Die Pixel des Integralbildes haben als Wert die Summe der Bildin-
tensitäten in dem Rechteck von der linken oberen Ecke bis zur eigenen Position. Einmal
berechnet, lässt sich die Summe der Intensitäten eines beliebig großen Rechtecks mit nur
zwei Addition und zwei Subtraktionen berechnen, indem man die Integralwerte der Diago-
nale addiert und die der Gegendiagonale subtrahiert. Besonders bei glättenden Verfahren
unter Verwendung mehrerer Skalierungen (*Skalen-Pyramide*) wird eine hohe Effizienzsteige-
rung erreicht. So werden nicht mehrere, verschieden skalierte Intensitätenbilder berechnet,
sondern die Glättungsfilter an sich skaliert, bei gleich bleibender, konstanter Rechenzeit.
Bay *et al.* konnten daraus ein BK-Verfahren (*SURF* [2]) ableiten, welches ähnlich dem
SIFT-Verfahren [7] ist, jedoch um den Faktor drei schneller.

Bei der Berechnung des Integralbildes durch die GPU ergibt sich die Schwierigkeit, dass
die Integration nur bedingt zur Parallelisierung geeignet ist. Es zeigt sich aber, dass eine
Bildzeilen-Integration gefolgt von einer Bildspalten-Integration ein korrektes Integralbild
erzeugt. Dadurch ergeben sich voneinander unabhängige Bildbereiche (zunächst Zeilen,
dann Spalten), auf welche die parallelen Verarbeitungsfäden (*Threads*) der GPU verteilt
werden können.

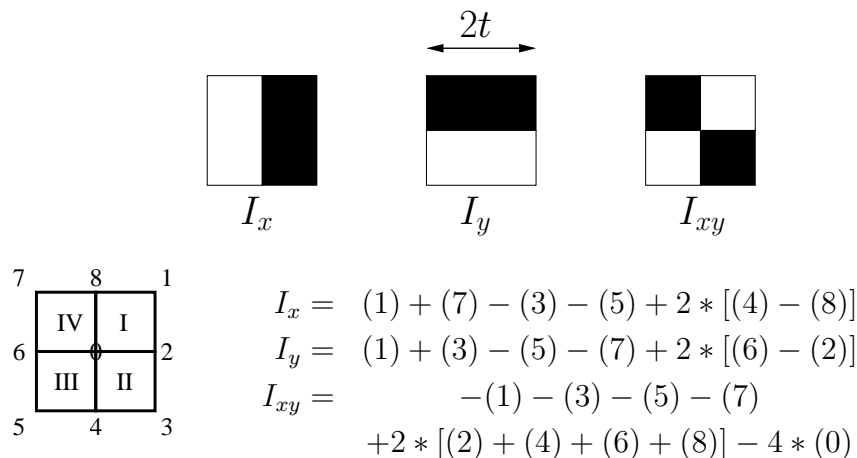


Abbildung 2. Komponenten des SidCell-Bildes und deren Berechnung mittels Integralbild.

Das SidCell-Bild $S_t(x, y)$ einer Skalierung t besteht aus dreidimensionalen Pixeln mit den
Filterantworten $(I_x, I_y, I_{xy})_t^\top$ als Komponenten. Abbildung 2 zeigt die Struktur der ein-
zelnen Filter, wobei I_x und I_y Differentialfilter 1. Ordnung sind. I_{xy} approximiert die
Krümmung des Verlaufes der Intensitätenfunktion in x - und y -Richtung (*Verdrillung* [5]).

Die Berechnung eines Pixels von S_t beschränkt sich damit auf 18 Additionen, 4 Multiplikationen und 9 Speicherzugriffe. Weitere 3 Divisionen kommen durch die Normierung der einzelnen Komponenten auf das Intervall $[-1, 1]$ hinzu.

2.2. Extraktion von Bildschlüsselpunkten

Es zeigt sich, dass der I_{xy} -Filter lokale Extrema an Stellen erzeugt, welche Gradienten in x - und y -Richtung haben. Idealerweise ist dies gegeben bei Ecken- sowie Schachbrettmuster parallel zu den xy -Achsen. Andererseits werden solche Strukturen bei einer Verdrehung um 45° nicht erkannt, woraus sich eine Rotationsvarianz ergibt. Was zunächst als nachteilig erscheint, ist auf den zweiten Blick weniger relevant. Die Rollbewegung eines KFZs kann in dieser Hinsicht vernachlässigt werden. Die Implementierung einer Rotationsinvarianz ist immer mit zusätzlichem Rechenaufwand verbunden, welcher hier eingespart wird. Skalierungsinvarianz jedoch ist ein wichtiges Merkmal für automotives Sehen. Diese ist durch den Skalenparameter t gegeben. Auf die automatische Skalenraumlokalisierung eines Keypoints wird hier nicht eingegangen.

Die Extraktion wird nun mittels einer *Non-Maximum-Suppression (NMS)* auf den Absolutbetrag der I_{xy} -Komponente durchgeführt. Die NMS vergleicht jeden $|I_{xy}|$ -Wert mit seinen acht Nachbarwerten. Alle Pixel, welche in dieser Hinsicht nicht maximal gegenüber ihrer Nachbarschaft sind, werden *unterdrückt*, sprich scheiden als Keypoint-Kandidaten aus. Zusätzlich wird eine Rauschschwelle $|I_{xy}| \geq |I_{xy}|_{\min}$ angewendet. Pixel, welche unter diese Schwelle fallen, werden ebenfalls unterdrückt. Abbildung 3 zeigt Ergebnisse dieser Extraktion bei verschiedenen Schwellwerten.

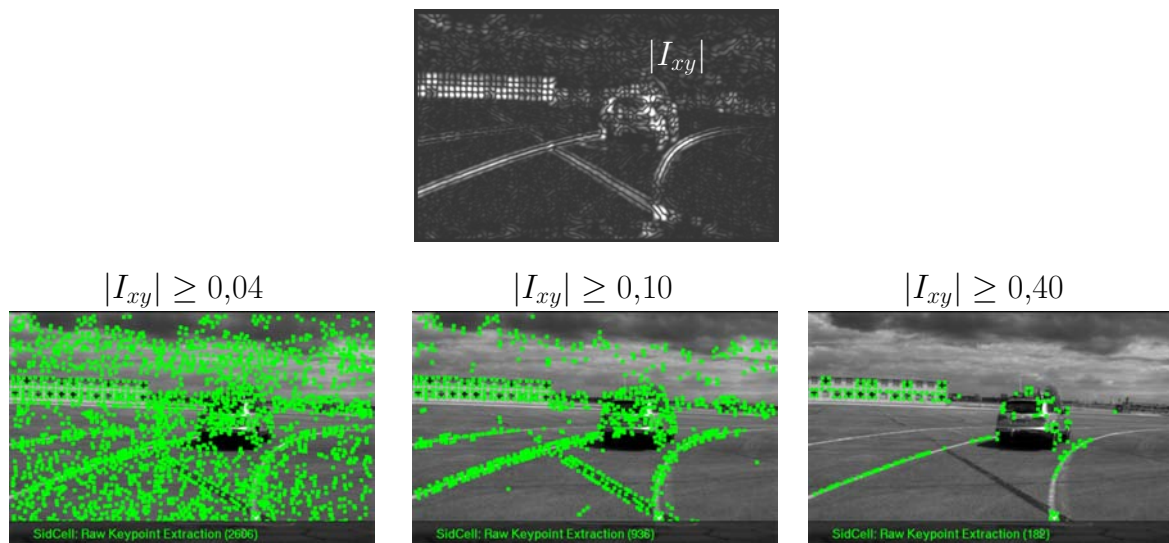


Abbildung 3. SidCell Bildschlüsselpunkt Extraktion durch NMS mit Schwellwert.

An dieser Stelle wollen wir genauer auf die GPU-Implementierung eingehen. Nach Abschluss der NMS, welche lokal unabhängig und damit stark parallelisierbar ist, liegt ein *binäres* Speicherbild vor. Jedem Bit entspricht *ein* Pixel und wird auf eins gesetzt, wenn

das Pixel nicht unterdrückt wurde. Zur weiteren, parallelisierten Verarbeitung ist ein solches, *dünn besetztes* Speicherbild nicht geeignet. Erst eine *dichte* Liste von Schlüsselpunkten würde wieder volle Vektorisierung erlauben. Meist wird hier ein sequentieller CPU-Schritt eingefügt, um die Schlüsselpunkte aus dem NMS-Speicherbild aufzusammeln und in eine Liste umzuwandeln. Die Liste wird dann wieder auf die GPU geladen. Solche Speichertransfers zwischen CPU und GPU können den Zugewinn an Rechengeschwindigkeit stark schmälern, oder gar komplett aufbrauchen. In [10, 11] zeigen wir, wie mittels atomarem Inkrementieren störende CPU-Schritte gänzlich vermieden werden können.

2.3. Bewertung von Bildschlüsselpunkten

Ein Problem des $|I_{xy}|$ -Detektors ist, dass entlang 45° -Kanten und -Streifen sehr viele falsch-positive Schlüsselpunkte auftreten. Solche Punkte weisen eine ungenügende Wiederholbarkeit aufgrund stark rauschender Lokalisierung auf. Zu sehen ist dies in Abbildung 3 bei den Spurmarkierungen und dem schwarzen Pitumenstreifen. Genau genommen handelt es sich um zwei verschiedene Arten von Falschdetektionen, wie Abbildung 4(ab) zeigt. Man kann leicht erkennen, dass ein $|I_{xy}|$ -Filter halber Skalierung (rot) in den Fällen 4(ab) einen Wert nahe Null liefert. Damit besteht eine starke Diskrepanz zum Original-Filter. Diese Diskrepanz besteht in den richtig positiven Fällen 4(cd) nicht, wodurch die Fälle 4(ab) und 4(cd) unterscheidbar werden.

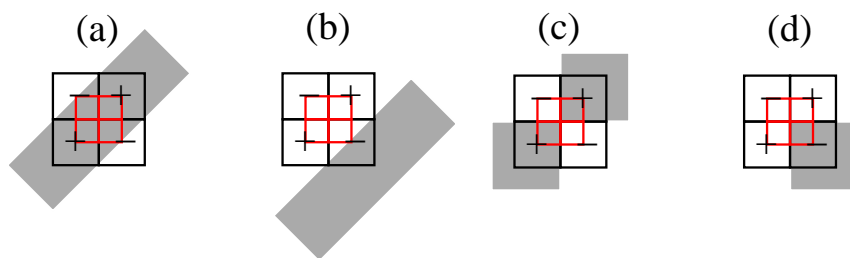


Abbildung 4. Falsch positive Schlüsselpunkte in (ab) und richtig positive in (cd). Die Unterscheidung kann mit einem $|I_{xy}|$ -Filter halber Skalierung (rot) getroffen werden.

4(a) und 4(b) kann man ebenfalls unterscheiden, da in 4(a) starke Diskrepanz zu den Filtern $|I_x|, |I_y|$ vorherrscht ($I_x, I_y \approx 0$), wogegen in 4(b) $|I_{xy}| \approx |I_x| \approx |I_y|$ gilt. Selbe Beziehung gilt für 4(c) und 4(d). Abbildung 5 zeigt eine solche Bewertung von Schlüsselpunkten.

2.4. Klassifizierung von richtig positiven Schlüsselpunkten

Wie in Abbildung 6 zu sehen, kann man Schlüsselpunkte, welche als richtig positiv bewertet wurden, wiederum in zehn verschiedene Klassen unterteilen. Dabei wird die Vorzeichen-Signatur des SidCell-Pixels sowie das Verhältnis der I_{xy} -Komponente zu den I_x, I_y -Komponenten herangezogen. Im nächsten Abschnitt zeigen wir, wie mittels dieser Klassifizierung die Korrespondenzberechnung effizienter durchgeführt werden kann.

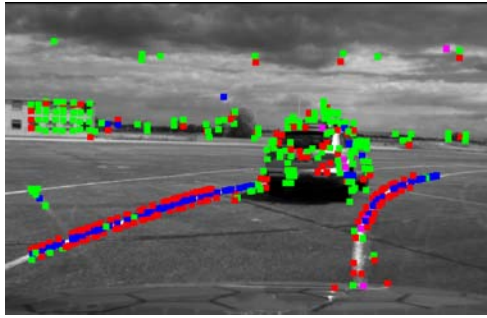


Abbildung 5. Falsch positive Schlüsselpunkte 4(a) sind blau oder 4(b) rot und richtig positive 4(c,d) grün dargestellt.

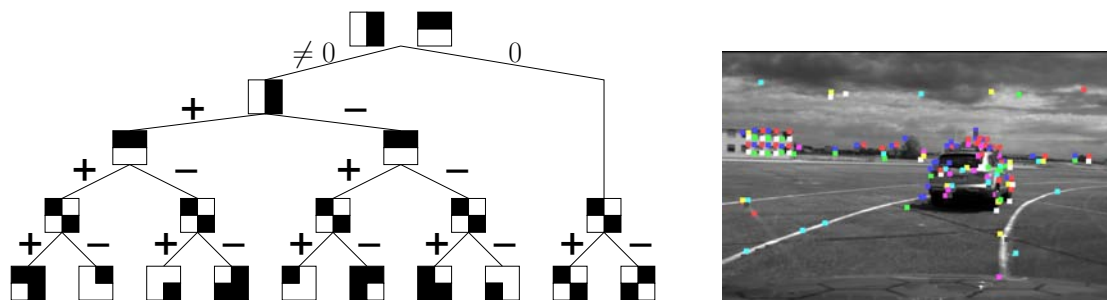


Abbildung 6. Klassifizierung in zehn verschiedene Klassen von Schlüsselpunkten (siehe Blätter des Baumes links). Im rechten Bild sind richtig positive Schlüsselpunkte aufgrund ihrer Klassenzugehörigkeit eingefärbt.

2.5. Korrespondenzberechnung im automotiven Bildstrom (BK)

Der Bildstrom einer in diesem Falle automotiven Kamera weist wichtige Merkmale auf. Zum einen liegt die Bildfrequenz in einem Bereich ≥ 20 Hz. Dies hängt in erster Linie davon ab, welche Geschwindigkeiten in der zu erkennenden Szene auftreten. Eigene Experimente zeigen, dass maschinelles, visuelles Spurhalten bei einer Geschwindigkeit von 40 km/h und engen Kurven bei einer Bildfrequenz von 20 Hz machbar ist. Ein weiteres Merkmal ist, wie schon erwähnt, die Vernachlässigbarkeit der Rollkomponente. Schliesslich kann die Eigenbewegung noch durch ein planares Bewegungsmodell mit lokal zirkularer Charakteristik angenähert werden [8, 9, 10]. Aus diesen Eigenschaften kann man wertvolle Vereinfachungen für den BK-Prozess ableiten. Aufgrund der hohen Bildfrequenz ist der Bildversatz eines Schlüsselpunktes eingrenzbar. Ein *jeder-mit-jedem* Abgleich (Komplexität $O(n^2)$, bei n Schlüsselpunkten) kann also auf eine Umgebung des Schlüsselpunktes reduziert werden. Die stabile Rolllage bewirkt, dass die Schlüsselpunkte nicht ihre Klasse (siehe voriger Abschnitt) wechseln. Das planare Eigenbewegungsmodell ermöglicht dann eine genauere Prädiktion des Versatzes, was den Suchbereich verschiebt und weiter eingrenzt. Wir gehen darauf später genauer ein. Zunächst betrachten wir den BK-Prozess ohne Eigenbewegungsmodell und nehmen nur an, dass ein Schlüsselpunkt seine Klasse nicht wechselt und einen definierten Maximalversatz aufweist.

Häufig stützt sich der BK-Prozess auf sogenannte *Bilddeskriptoren*, welche die Umgebung

eines Schlüsselpunktes derart beschreiben, dass eine ausreichende Stabilität zur Wiedererkennung gegeben ist. Solche Deskriptoren sind meist hochdimensional (32-, 64-, bis zu 128-dimensional [7]). Berechnung und Vergleich sind dementsprechend aufwändig. In diesem Verfahren sollen *schwergewichtige* Deskriptorenvergleiche weitgehend vermieden werden. Die bereits erläuterte Klassenstabilität erlaubt, Deskriptorenvergleiche nur innerhalb der Klasse durchführen zu müssen. Weiterhin werden die Deskriptoren aus dem bereits berechneten Sidcell-Bild zusammengesetzt, wodurch kein zusätzlicher Rechenaufwand anfällt. Die Praxis hat gezeigt, dass in der Regel 32-dimensionale Deskriptoren für einen Bildstrom mit geringem Versatz ausreichen. Abbildung 7 verdeutlicht das *Matching*-Verfahren noch einmal. Aus den Bild-zu-Bild-Korrespondenzen (*f2f* für *frame to frame*) werden dann über der Zeit Langzeitkorrespondenzen (*lt* für *long term*) aufgebaut (siehe Abbildung 1).

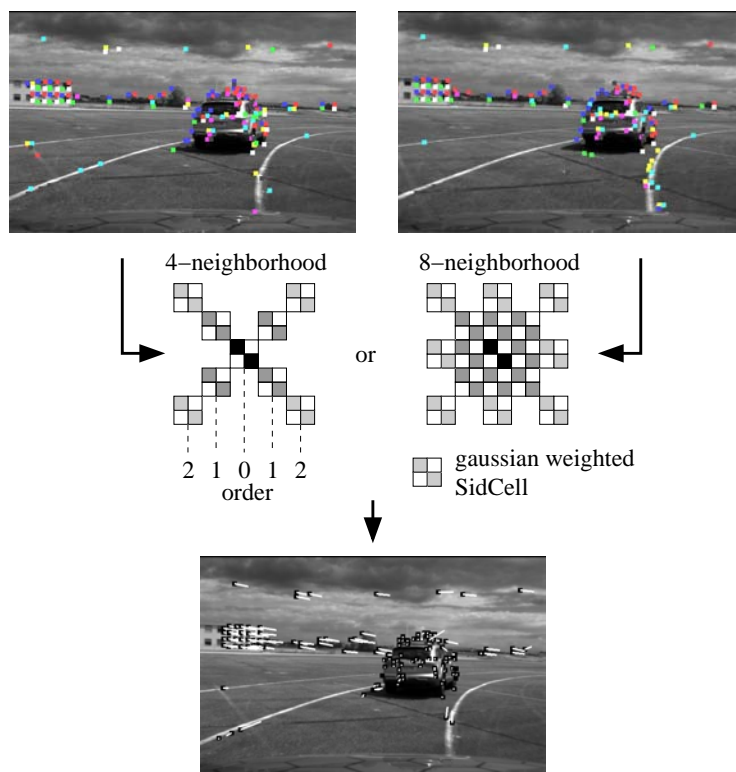


Abbildung 7. Matching von SidCell-Keypoints. In zwei aufeinander folgenden Bildern eines Kamerastroms werden nur Keypoints gleicher Klassen (eingefärbt) verglichen. Die Deskriptoren werden aus dem bestehenden SidCell-Bild durch Gauss-Gewichtung von benachbarten SidCell-Pixeln (4er- oder 8er-Nachbarschaft) zusammengesetzt.

2.6. Leistungsauswertung des SidCell BK-Prozesses

Bei der Auswertung der benötigten GPU-Zeit zeigt sich eine Laufzeit von unter 3ms inklusive Speichertransfer (siehe Tabelle 1 und Abbildung 8). Die Laufzeit ist im wesentlichen unabhängig von der Anzahl der extrahierten Schlüsselpunkte. Integralbild, SidCell-Bild, NMS und Speichertransfer benötigen ca. 80% der Gesamtzeit. Gerade beim Integralbild und

$ I_{xy} \geq$	0.05	0.10	0.15	0.20	0.30	0.40	0.50	0.60
\varnothing Keypoints [#]	1611	894	593	464	277	187	128	88
\varnothing GPU Zeit [ms]	2.75	2.60	2.53	2.49	2.45	2.41	2.38	2.35
\varnothing BK [#]	985	395	230	162	101	65	43	26
\varnothing BK Error [#]	156	31	16	6	4	3	2	0
\varnothing BK Error [%]	15.9	7.9	7.0	3.7	4.0	4.7	4.7	0.0

Tabelle 1. Auswertung über Laufzeit und Fehlerverhalten.

der NMS sehen wir noch großes Optimierungspotential. Erst danach werden die Schlüsselwortlisten verarbeitet, welche aber nur einen geringen Anteil an der Gesamtzeit haben. Bei der Fehlerquote der ermittelten BKs zeigt sich ab einer Rauschunterdrückung von 0.10 ein Fehler von unter 10%. Es muss darauf hingewiesen sein, dass die Fehlerrate stark von der Szene abhängt. In diesem Fall wurde eine typische automotiv Szene mit vorausfahrendem Fahrzeug und guten Lichtverhältnissen über einen längeren Zeitraum *per Hand* ausgewertet. Bei unkooperativen Szenen (Licht, Spiegelungen, schwache Texturierung oder Bewegungsunschärfe) können weitaus schlechtere Werte auftreten.

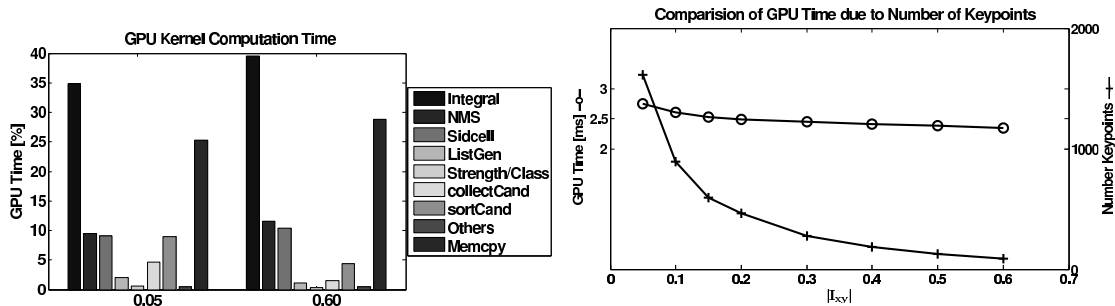


Abbildung 8. Laufzeitauswertung des SidCell BK-Prozesses. Die x -Achsen der beiden Graphen zeigen den $|I_{xy}|$ -Schwellwert an und damit implizit die Anzahl der verarbeiteten Schlüsselwörter.

Die Auswertung wurde mit einem 768×480 -Format durchgeführt. Die eingesetzte GPU war eine NVidia Tesla C1060.

3. Beispiele für die Anwendung in automotivem Sehen

In diesem Abschnitt wollen wir kurz auf Anwendungsbeispiele eingehen, wie sie in unserem Versuchsträger MuCAR-3 zu Einsatz kommen. MuCAR-3 ist ein umgerüsteter VW Touareg mit einem 8-Kern Intel Rechner und einer NVidia GTX 260 Grafikkarte. Die Aktorik ist vollständig über den Rechner steuerbar, als Sensorik dient unter anderem eine Kameraplattform mit 2 Weitwinkel- und einer inertial stabilisierten Telekamera.

3.1. Visuelle Odometrie

Die Kenntnis der Eigenbewegung spielt eine wichtige Rolle in der automotiven Bildverarbeitung. Während die Bahngeschwindigkeit durch den Tachometer bereits verfügbar ist,

sind die Rotationsgeschwindigkeiten schwerer mit Standard-Bordsensoren zu ermitteln. Möglich wäre ein Auslesen der ESP-Sensoren für ein Gierraten-Signal und die Stoßdämpferstellung für Nickraten- und Rollraten-Signal. Diese Signale können dann zum einen zur Stabilisierung des BK-Prozesses verwendet werden, oder auch zur Erkennung von nicht zur Eigenbewegung passenden, also dynamischen Objekten. Basierend auf der Arbeit in [10] wollen wir zeigen, wie diese Signale aus einem monokularem Bildstrom *ohne* externe Sensoren ermittelt werden können. Wichtige Konzepte dabei sind die *inverse Tiefenparametrierung IDP* [3] sowie eine verallgemeinerte Form der *Disparitätengleichung* (1) aus [6].

$$\tilde{\mathbf{m}}_1 = \mathbf{A}\mathbf{R}\mathbf{A}^{-1} \cdot \tilde{\mathbf{m}}_0 + X_0^{-1}\mathbf{A}\mathbf{t} \quad (1)$$

Diese Form der Disparitätengleichung ist günstig, da die Auftrennung von Rotations- und Translationsbewegung deutlich ersichtlich ist und der Strukturparameter X^{-1} der inversen Tiefe vorhanden ist. Der linke Teil der Rotation stellt eine Homographie dar und ist demnach eine *lineare* Abbildung. Diese Homographie $\mathbf{A}\mathbf{R}\mathbf{A}^{-1}$ wird auch als *Homographie im Unendlichen \mathbf{H}_∞* bezeichnet. Das Vorhandensein des rechten Terms macht die Gleichung dann *nicht-linear*. Dabei ist \mathbf{A} die intrinsische Kalibriermatrix der Kamera, \mathbf{R} die Rotationsmatrix, \mathbf{t} der Translationsvektor und $\tilde{\mathbf{m}}_0, \tilde{\mathbf{m}}_1$ die homogene Darstellung der BK. X_0 ist die Tiefe des 3D-Punktes zu \mathbf{m}_0 in Kamera 0. Diese Gleichung vereinfacht sich wesentlich, wenn man die bereits erwähnten Eigenschaften des automotiven Bildstroms anwendet. \mathbf{R} kann durch eine Kleinwinkelnäherung approximiert werden, und \mathbf{t} nach [9] durch den halben Gierwinkel Ψ und der Kreisbogensehne ρ , siehe Abbildung 9 links. Nach der Vereinfachung zeigt sich deutlich, dass das Vorhandensein eines kombinierten Struktur-Bewegungs-Parameters ρX^{-1} den optischen Fluss der reinen Rotationsbewegung *skaliert* und bei Nicht-Vorhandensein ($\rho = 0$ und/oder $X^{-1} = 0$) unverändert lässt. Da aus

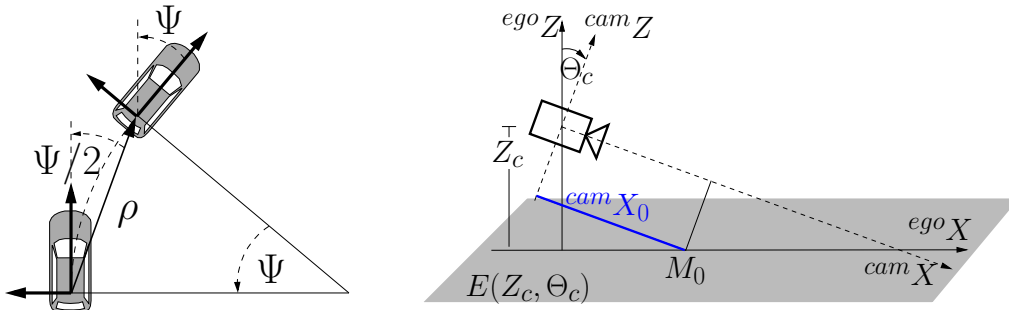


Abbildung 9. *Links:* nicht holonome, lokal zirkulare Bewegung eines Kfz's nach [9]. *Rechts:* Tiefe X_0 eines 3D-Punktes M_0 in den Koordinaten einer genickten Kamera (Winkel Θ_c) über Grund (Höhe Z_c).

dem monokularen Bildstrom direkt keine Tiefeninformation zu ermitteln ist, muss X^{-1} anderweitig bestimmt werden, z.B. durch ein *planares Tiefenmodell* wie in Abbildung 9 rechts. Hier wird der Vorteil der Verwendung der inversen Tiefe deutlich. Die Singularität der Unendlichkeit wird in den Ursprung der Kamera verlegt, welcher nicht im Bild sichtbar ist. Der Übergang von Punkten nahe dem Horizont zu Punkten im Unendlichen ist ein stetiger Übergang nach Null. Zudem ist die Abbildung von Bildzeile zu inverser Tiefe linear.

Nun kann man ein Kalman-Filter formulieren, um die vier Bewegungssignale rekursiv aus den SidCell-Korrespondenzen zu schätzen [10]. Abbildung 10 zeigt experimentelle Ergebnisse. Die Prädiktion des Filter wird verwendet, um über Gleichung (1) den BK-Prozess zu stützen.

3.2. Bewegungsstereo

Integriert man die Bewegungssignale über der Zeit, so kann man für lt-Korrespondenzen eine *Disparität* zur planaren Eigenbewegung berechnen. Diese Disparität wird durch Abweichung vom planaren Strukturmodell oder durch bewegte Objekte hervorgerufen. Abbildung 11 zeigt auch hierzu wieder experimentelle Ergebnisse.

3.3. Textur- und Modelltracking

Abbildung 11 zeigt, wie Bildbereiche segmentiert werden können, die eine Eigenbewegung aufweisen, z.B. vorausfahrende Fahrzeuge. Zu diesem Zeitpunkt liegt noch keine Vorstellung über die Art des möglicherweise erkannten Objekts vor, es handelt sich nur um eine Hypothese. Somit kann noch kein modellbasiertes Tracking erfolgen. Um aber den identifizierten Bildbereich über einen längeren Zeitraum zu analysieren und schliesslich einem Modell zuzuschreiben, wird basierend auf SidCell-Schlüsselpunkten und Deskriptoren ein Textur-Tracking durchgeführt. Wichtig ist, dass nun auch die Skalierung fortlaufend angepasst wird, um bei einer Entfernungsänderung die Textur nicht zu verlieren. Abbildung 12 zeigt dies am Beispiel einer Rückansichttextur.

Modellbasiertes Tracking kann durchgeführt werden, indem 2D-SidCell-Merkmale in ein 3D-Objektmodell eingetragen und wiedererkannt werden. Der algorithmische Aufbau solcher Modelle ist Gegenstand aktueller Arbeit. Ist ein Merkmalsmodell einmal einem Bildbereich zugeordnet, kann die 6D-Pose des Objekt fortlaufend geschätzt werden. Im realen Experiment wurden damit Konvoi-Szenarien bewältigt. Abbildung 13 zeigt hierzu Ergebnisse.

4. Zusammenfassung und Ausblick

Wir haben ein echtzeitfähiges Verfahren zur Korrespondenz-Berechnung vorgestellt. Mit Spezialisierung auf den automotiven Bildstrom werden GPU-Zeiten unter 3ms erreicht. Aus den Korrespondenzen konnte die Eigenbewegung in hoher Güte geschätzt werden. Ein Abgleich mit einem planaren Tiefenmodell erzeugt Hypothesen für bewegte und unbewegte Objekte, wie z.B. andere Fahrzeuge. Aktuelle und zukünftige Arbeit beschäftigt sich mit der Gruppierung und 3D-Modellierung von SidCell-Korrespondenzen aus Informationen wie z.B. durch Bewegungsstereo.

Literatur

- [1] A. Barth and U. Franke. Where will the oncoming vehicle be the next second? In *Proc. IEEE Intelligent Vehicles Symposium*, pages 1068–1073, Jun. 2008.
- [2] H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded up Robust Features. In *Proc. European Conf. on Computer Vision*, 2006.

- [3] J. Civera, A. Davison, and J. Montiel. Inverse Depth Parametrization for Monocular SLAM. *IEEE Transactions on Robotics*, 24(5):932–945, Oct. 2008.
- [4] T. Dang. *Kontinuierliche Selbstkalibrierung von Stereokameras*. Dissertation, Universität Karlsruhe, MRT, 2007.
- [5] E. D. Dickmanns. Corner Detection with Minimal Effort on Multiple Scales. In *Proc. Int'l Conf. on Computer Vision Theory and Applications*, pages 315–320, 2008.
- [6] R. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, 2006.
- [7] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int'l J. of Computer Vision*, 2004.
- [8] D. Scaramuzza, F. Fraundorfer, M. Pollefeys, and R. Siegwart. Absolute Scale in Structure from Motion from a Single Vehicle Mounted Camera by Exploiting Nonholonomic Constraints. In *Proc. IEEE Int'l Conf. on Computer Vision*, Oct 2009.
- [9] D. Scaramuzza, F. Fraundorfer, and R. Siegwart. Real-Time Monocular Visual Odometry for On-Road Vehicles with 1-Point RANSAC. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, May 2009.
- [10] M. Schweitzer, A. Unterholzner, and H.-J. Wuensche. Real-Time Visual Odometry for Ground Moving Robots using GPUs. In *Proc. Int'l Conf. on Computer Vision Theory and Applications*, May 2010. *accepted, full paper*.
- [11] M. Schweitzer and H.-J. Wuensche. Efficient Keypoint Matching for Robot Vision using GPUs. In *Proc. 12th IEEE Int'l Conf. on Computer Vision, 5th IEEE Workshop on Embedded Computer Vision*, Oct. 2009.
- [12] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.
- [13] Web-Video. <http://youtube.com/watch?v=OFfynOWVkwg>.

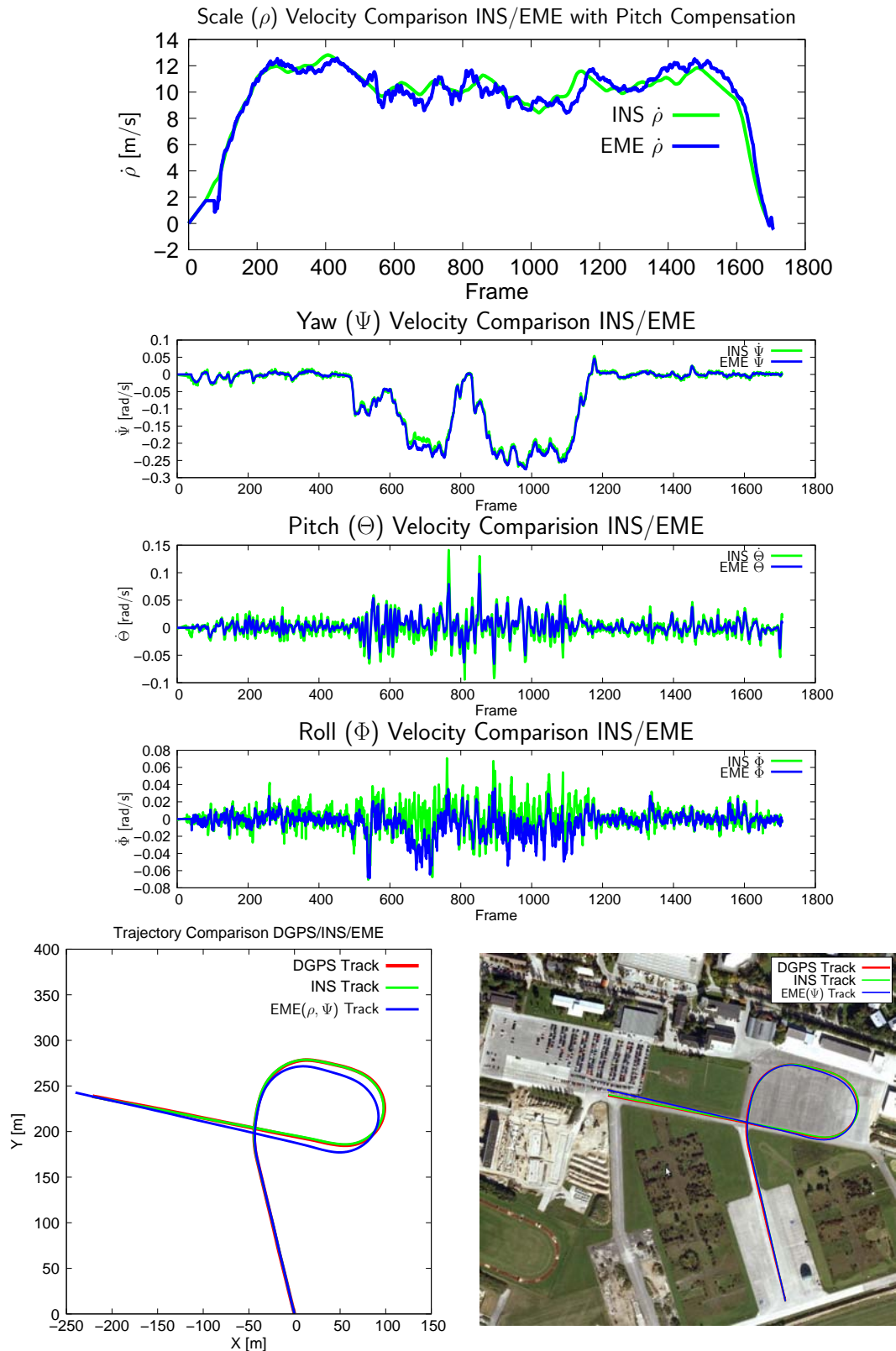


Abbildung 10. Bewegungssignale durch visuelle Odometrie aus einem monokularen Bildstrom (blau, Ground Truth durch INS grün).

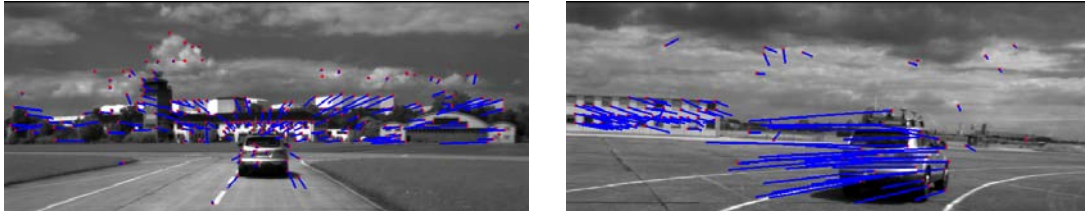


Abbildung 11. Bewegungsstereo durch Abweichung vom planaren Tiefenmodell.

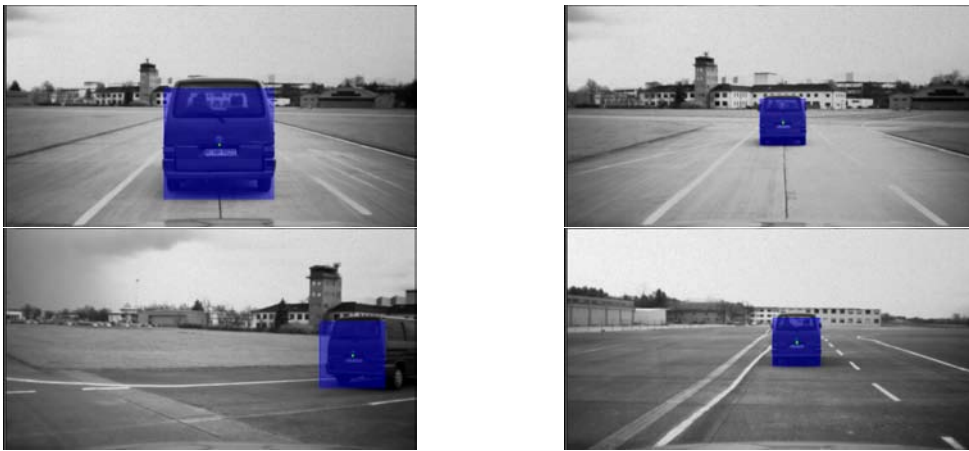


Abbildung 12. Skalierungsinvariantes Texturtracking einer Rückansicht.



Abbildung 13. Tracking von 3D-Merkmalenmodellen. Links zeigt die rote Linie den prädizierten weiteren Trajektorienverlauf aufgrund des Einspurmodells. Rechts ist die Referenzierung der Posenschätzung (grünes Modell) zu DPGS Ground Truth (blaues Modell) dargestellt.