

Fügen Sie nun ein Empty hinzu und geben Sie ihm im Physics Context ein Kraftfeld durch den Button Force Field. Stellen Sie den Typ auf Wind und richten Sie das Empty so aus, dass dem Fellball der Wind die Haare zerzaust. Stärke und Richtung können Sie wie gewohnt interaktiv während des Abspielens einstellen.

8.6 Physikalische Animation

Neben der Beeinflussung von Partikelsystemen durch Felder gibt es in Blender noch das weite Feld der physikalischen Animationen wie z. B. die Simulation von Rauch, Flüssigkeiten und Kleidung. Teilweise haben diese Verfahren Gemeinsamkeiten mit den Partikelsystemen, benutzen Partikel als Steuerelemente oder werden durch die gleichen physikalischen Effekte beeinflusst.

8.6.1 Cloth

Die Kleidungssimulation von Blender können Sie als Modellierungstool und natürlich auch als Animationstool benutzen. Viele Dinge wie Tücher, Tischdecken, aber auch Kleidung für Characters, die später nur auf Standbildern auftauchen sollen, sind so machbar. Für die Animation wiederum verformt sich die Kleidung mit den Bewegungen des Characters, Fahnen flattern im Wind und Seile bewegen sich realistisch. So einfach die grundlegende Bedienung des Cloth-Simulators ist, so komplex kann die Verwendung von Kleidungssimulationen in aufwändigen Szenen werden.

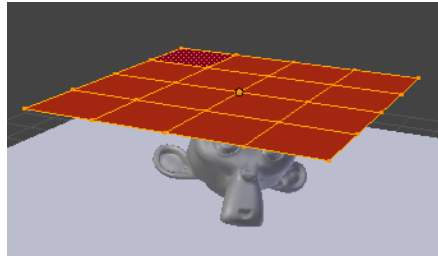


Abb. 8–22

Szene mit Cloth- und Fluid-Simulation



Übung!



Erstellen Sie eine Ebene, die als Boden dient, und darauf ein einfaches Grundobjekt oder auch den Affenkopf (Cloth/Samtuch00.blend). Über dem Objekt platzieren Sie eine weitere Fläche, die uns als Testobjekt für die Cloth-Simulation dienen wird. Wechseln Sie für

das Testobjekt (ein Tuch) in den Edit Mode und unterteilen Sie die Fläche mittels **W** → Subdivide zwei Mal, verlassen Sie dann den Edit Mode.

Wechseln Sie in den Physics Context und fügen Sie dem Tuch durch Klick auf den Cloth Button eine Cloth-Simulation hinzu. Wenn Sie nun im 3D View mit **Alt**-**A** die Animation abspielen, so sehen Sie das Tuch herunterfallen – noch fällt es allerdings durch unsere weiteren Objekte hindurch.

Selektieren Sie nun den Affenkopf (oder Ihr Objekt) und aktivieren Sie Collision im Physics Context. Das Gleiche machen Sie auch mit der Bodenfläche. Nun wird das Tuch beim Abspielen der Animation die Objekte nicht mehr durchdringen.

Wirklich realistisch sieht die Simulation aber noch nicht aus, dazu ist das Tuch noch zu grob aufgelöst. Wir könnten jetzt mit **W** → Subdivide weitere Unterteilungen hinzufügen, allerdings ist dieses Verfahren sehr unflexibel. Besser ist es, einen Subdivision Surface Modifier zu verwenden. Selektieren Sie also das Tuch und erstellen Sie im Modifier Context einen Subdivision Surface Modifier. Setzen Sie View: und Render: vorerst auf 2 und schalten Sie im Tool Shelf das Shading: auf Smooth. Spielen Sie jetzt mit **Alt**-**A** die Animation nochmals ab: Die Fläche sieht besser aus, verhält sich aber immer noch nicht korrekt. Dies liegt, wie Sie sicher schon ahnen, an der Position des Subdivision Surface Modifier auf dem Modifier Stack. Die Cloth Simulation wird nämlich vor dem Subdivision Surface Modifier ausgeführt und arbeitet damit natürlich noch mit den Originaldaten. Also rücken Sie den Subdivision Surface Modifier mittels der kleinen Pfeile über den Cloth Modifier. Möchten Sie die Ecken des Tuchs nicht verrundet bekommen, so wählen Sie noch Simple im Subdivision Surface Modifier. Die Animation sollte nun wesentlich realistischer aussehen. Sie können jetzt durch Erhöhen der Subdivisions: für View: und Render: testen, ab wann Ihr Rechner in die Knie geht. Bei jeder Änderung sollten Sie aber mit **Ctrl**-**Home** an den Anfang der Animation springen.

Vielleicht haben Sie bemerkt, dass der erste Durchlauf länger dauert und weitere Durchläufe dann in Echtzeit ablaufen. Das liegt daran, dass Blender für physikalische Simulationen einen sogenannten Cache, also einen Zwischenspeicher, erstellt. Auch wenn Sie die Animationszeit über die Timeline oder den Render Context verlängern, wird die Cloth-Animation nur 250 Bilder lang abgespielt. Der Grund liegt im Cloth Cache-Panel der Cloth-

Cloth Simulation



Kollisionsobjekte



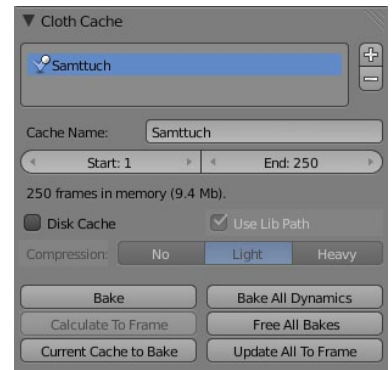
Cloth Cache

Simulation im Physics Context: Hier ist End: mit 250 angegeben. Erhöhen Sie End:, falls die Simulation bei Frame 250 noch nicht zu einem Stillstand gekommen sein sollte. Im Prinzip haben wir schon bei den Partikeln mit Caches gearbeitet, allerdings mussten wir hier die Cache-Länge nicht manuell anpassen.

Die Caches in Blender können entweder mit der Szene gespeichert werden oder per Disk-Cache auf der Festplatte, was sich bei komplexeren Simulationen anbietet, damit der Hauptspeicher für die eigentliche Szene frei bleibt. Der Cache wird bei jedem ersten Abspielen der Animation aktualisiert, sofern Parameter in der Cloth-Simulation geändert wurden. Durch Bake wird explizit die Simulation in den Cache »gebacken« und danach die Simulation gegen versehentliches Ändern gesperrt. Gleiches leistet Current Cache to Bake, das die beim Abspielen der Animation berechnete Simulation benutzt, um den Cache zu füllen. Durch Free Bake wird der Cache und damit die Simulation wieder für Änderungen frei gegeben.

Wahrscheinlich haben Sie schon die Fehler in der Simulation bemerkt: Der Stoff durchdringt sich in Falten selbst und auch ein Erhöhen der Auflösung ändert daran nichts. Und wenn Sie das Tuch nicht ganz zentrisch auf den Affenkopf fallen lassen, so rutscht es herunter, weil es keine Reibung zwischen Tuch und Kopf bzw. Boden gibt.

Die Selbstdurchdringung können wir sehr schnell durch Aktivieren der Option Self Collision im Cloth Collision-Panel des Physics Context abschalten, jetzt versucht die Simulation solche Selbstdurchdringungen zu verhindern. Schlägt das einmal fehl, so kann man über die Quality:- und Distance:-Parameter noch Einfluss auf die Berechnung nehmen. Ähnliches ist im Cloth Collision-Panel auch für die Kollisionen von Stoffen mit anderen Objekten einstellbar und für diesen Fall zusätzlich noch der Parameter Friction: (Reibung), der auch unser letztes Problem löst, das zu starke Gleiten des Stoffes. Erhöhen Sie also Friction:, bis das Ergebnis Sie überzeugt.



Die Simulation verfeinern

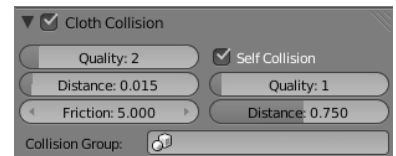


Abb. 8–23

Was versteckt sich da wohl?



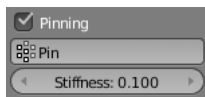
Im Cloth-Panel können Sie jetzt noch weitere Parameter ändern. Ganz prominent sind hier die Presets:, wohinter sich Voreinstellungen für verschiedene Stoffe verstecken. So verhält sich natürlich Seide (Silk) ganz anders als Jeansstoff (Denim). Testen Sie die verschiedenen Stoffe und schauen Sie sich die Parameter an, um herauszufinden, was sie bewirken.

Mittels Pre roll: können Sie übrigens die Simulation schon eine Anzahl von Frames: vorlaufen lassen und so die Cloth-Simulation bereits auf einen stabilen Zustand hinarbeiten lassen.

Sie können nun auch einmal mittels Keyframeanimation den zugedeckten Körper animieren oder ein Force Field wie z. B. Wind auf das Tuch wirken lassen. Bei der Keyframeanimation könnte es jetzt passieren, dass das zugedeckte Objekt bei schnellen Bewegungen durch das Tuch dringt. Dies beheben Sie durch Erhöhen von Quality: im Cloth Collision-Panel und eventuell Quality: im Cloth-Panel.

Pinning

Fallende Tücher sind ja ganz schön, aber oft muss man Stoff an etwas befestigen. Ein T-Shirt über einem Körper funktioniert prima mit der Cloth-Simulation, einen Vorhang kann man tatsächlich mit Löchern modellieren, in denen eine Gardinenstange läuft, die so den Vorhang festhält. Dies ist aber oft viel zu aufwändig. Daher gibt es die Möglichkeit, einzelne oder mehrere Vertices zu fixieren, das sogenannte »Pinning«. Und wie sollte es anders sein: Es funktioniert über Vertex Groups.



Erstellen Sie also eine Vertex Group, benennen Sie sie z. B. »Pin« und weisen Sie ihr die zu fixierenden Vertices mit dem Weight: von 1.000 zu. Natürlich lassen sich die Vertex-Gruppen auch wieder per Weight Paint verfeinern.

Im Cloth-Panel schalten Sie nun Pinning ein und wählen die Vertex Group »Pin« aus, schon sind die entsprechenden Vertices während der Simulation fixiert.

8.6.2 Smoke

Mit dem Smoke-Simulator in Blender ist es, wie der Name schon sagt, möglich, Rauch zu erzeugen, der sich sehr realistisch in einem Gebiet bewegt, mit Objekten interagiert und auf Kraftfelder reagiert. Sehr praktisch ist, dass man mit entsprechenden Texturen auch einfach das passende Feuer zum Rauch bekommt. Obwohl das initiale Setup für ein Rauch-System schnell erklärt ist, sind es doch wieder die Feinheiten, die später das Ergebnis ausmachen. So kann dieses kurze Tutorial nur die Basis für Ihre eigenen Experimente bilden. Aufbauend darauf und mithilfe von Tutorials aus Spezialbüchern und aus dem Internet können Sie dann Ihre Simulation verfeinern und für den konkreten Anwendungsfall erstellen.

Eine Rauchsimulation in Blender besteht immer aus einer Domain, in der die Simulation abläuft, und einem Rauchemitter (Flow). Auf dem Rauchemitter muss sich ein Partikelsystem befinden, dessen Partikel die Startpositionen und Geschwindigkeit des Rauchs bestimmen. Damit kann im 3D View schon der Rauch simuliert werden. Zum Rendern kommt allerdings noch ein Volumenmaterial inkl. einer Voxel-Data-Textur, in der die Ergebnisse der Rauchsimulation gespeichert sind, hinzu.

Was hat die arme Suzanne nicht schon alles erliden müssen. In der folgenden kleinen Übung werden wir sie auch noch in Flammen setzen (die Szenen finden Sie unter Smoke/SuzanneBMF00.blend usw.). Erstellen Sie also einen Affenkopf mit \square -[A]→Mesh →Monkey. Dann erzeugen Sie einen Würfel und skalieren ihn etwa um den Faktor 7. Dieser Würfel wird unsere Domain, in der die Rauchsimulation abläuft.

Skalieren Sie Domains immer im Object Mode und führen Sie später keinesfalls ein Apply Size mit Strg -[A] aus.

Platzieren Sie die Domain so, dass der Affenkopf etwa am Boden der Domain sitzt. Wechseln Sie bei selektiertem Würfel in den Physics Context und fügen Sie dem Würfel durch einen Klick auf den Smoke-Button eine Smoke-Simulation hinzu. Im Smoke-Panel ändern Sie dann den Typ auf Domain. Der wichtigste Parameter hier ist Divisions:, der die Auflösung der Rauchsimulation bestimmt. Für die ersten Versuche lassen wir Divisions: bei 32 stehen; dies reicht bei normalen Domaingrößen für eine noch schnell zu erstellende Vorschau aus. Schalten Sie jetzt noch Dissolve an und setzen Sie Time: auf 20. Dies sorgt dafür, dass sich der Rauch nach 20 Frames aufzulösen beginnt.

Selektieren Sie den Affenkopf und erstellen Sie im Particles Context ein neues Partikelsystem. Die Grundeinstellungen können im Großen und Ganzen übernommen werden. Setzen Sie im Emission-Panel die Lifetime: auf 6 und Random: auf 0.600. Schalten Sie No im Physics-Panel ein – im Moment brauchen wir zwar keine physikalische Berechnung der Partikel, bei komplexeren Setups kann es aber sein, dass wir den Partikeln eine Geschwindigkeit und Richtung geben müssen, z. B. bei einer Particle-Fluid-Simulation, um brennende Flüssigkeiten zu erzeugen. Schalten Sie im Render-Panel auf None, so dass die Partikel nicht berechnet werden. Damit wir sie trotzdem im 3D View sehen, schalten Sie im Display-Panel Point ein.

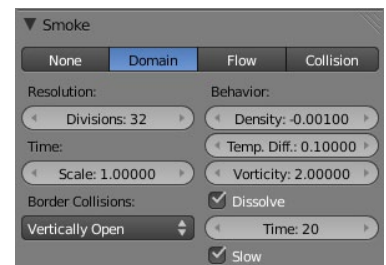
Bei weiterhin selektiertem Affenkopf wechseln Sie in den Physics Context, aktivieren hier Smoke und schalten den Typ auf Flow um. Im nun gefüllten Smoke-Panel ist schon das passende Partikelsystem eingetragen. Setzen Sie Temp. Diff: auf 2.000 hoch – dies sorgt für eine hohe Temperaturdifferenz zwischen Rauch und Umgebung, damit der Rauch schnell aufsteigt. In diesem Panel finden Sie auch die Option Initial Velocity, die wichtig

Grundlagen

Übung!



Domain



Partikelsystem

Rauchgenerator: Flow

wird, wenn Sie wie oben angesprochen die Partikel schon mit einer physikalischen Simulation versehen wollen.

Berechnung



Schalten Sie nun mit **[⇧]-[←]** die Animationszeit auf den ersten Frame und spielen Sie die Animation im 3D View mit **[Alt]-[A]** ab: Die Simulation wird berechnet und der Rauch bildet sich aus. Wenn die Animation von vorne anfängt, wird die Bewegung schneller, weil die Simulation nun aus dem Cache abgespielt wird. Wie bei der Cloth-Simulation spielt auch hier der Cache eine große Rolle. Wenn Sie die Simulation in der Länge anpassen wollen, so müssen Sie nicht nur die Animationszeit ändern, sondern auch immer die Start- und End-Werte im Smoke Cache-Panel des Domainobjekts. Sind Sie so weit mit dem Rauch zufrieden, können Sie die Simulation schützen, indem Sie **Current Cache to Bake** anwählen. Wenn Sie aufwändigere Simulationen, die durchaus Stunden zur Berechnung benötigen können, berechnen müssen, sollten Sie direkt den **Bake**-Button benutzen.

Material



Vielleicht haben Sie schon einmal **[F12]** gedrückt und nur eine graue Wand gesehen, wo doch im 3D View Rauch ist. Dies ist die Domain, die im Moment noch als normales Objekt gerendert wird. Selektieren Sie die Domain und erstellen Sie im Material Context ein neues Material. Der Typ des Materials muss auf **Volume** gesetzt werden.

Setzen Sie **Density**: auf 0.000 und **Density Scale**: auf 4.000. Dies bedeutet, dass die Dichte des Rauchs nur von der noch zu erstellenden Textur bestimmt und um den Faktor 4 dichter berechnet wird. Im **Shading**-Panel setzen Sie **Scattering**: auf 0.600 und **Emission**: auf 6.000 – dies sorgt für ein leuchtendes Feuer.

Ein wichtiger Wert bezüglich Rechenzeit und Qualität ist **Step Size**: im **Integration**-Panel. Setzen Sie ihn erst einmal auf 0.100. Wenn dann im Feuer (besonders am Emitter) Rauschen zu sehen sein sollte, so verkleinern Sie **Step Size**:. Dieser Wert beeinflusst allerdings besonders stark die Renderzeit von Volumenmaterialien.

Textur

Wechseln Sie in den **Texture** Context und erzeugen Sie eine neue Textur durch einen Klick auf **New**. Den Typ der Textur schalten Sie auf **Voxel Data** um und tragen unter **Domain Object** Ihr Domain-Objekt ein. Im **Influence**-Panel schalten Sie **Emission Color**: ab und **Density**: an.

Wenn Sie mit **[F12]** rendern, sehen Sie das erste Mal etwas von dem Rauch, durch die **Emission**:-Einstellung allerdings sehr überstrahlt. Würden Sie nur ein Rauchmaterial generieren wollen, so müssten Sie **Emission**: auf 0.000 setzen und wären im Prinzip fertig.

Feuer!

Erzeugen Sie eine weitere Textur im zweiten **Texture**-Slot, genau wie oben vom Typ **Voxel Data**, das **Domain Object** erstellen Sie wie gehabt. Im **Influence**-Panel schalten Sie **Emission Color**: und **Emission**: an, **Blend**: schalten Sie auf **Multiply**:. Jetzt moduliert die Rauchdichte die Emission und Dichte des ersten **Texture**-Slots.

Der wichtigste Teil für das Feuermaterial findet sich aber im **Colors**-Panel. Schalten Sie hier **Ramp** an und erstellen Sie einen Farbverlauf, wie in

der nebenstehenden Abbildung zu sehen. Diese Ramp bestimmt die Farbe, die für eine bestimmte Dichte in der von der Smoke-Simulation generierten Voxel-Data-Textur benutzt wird. Links ist der Bereich mit der geringsten Dichte, hier wird Rauch gerendert, rechts ist die höchste Dichte und damit das hellste Feuer bzw. die hellste Emission.

Je größer der Bereich links ist, desto mehr Rauch wird gerendert. Aber eventuell muss dann die *Dissolve: Time:* in der Domain erhöht werden, um auch noch Feuer zu erhalten, und/oder der Farbverlauf angepasst werden.

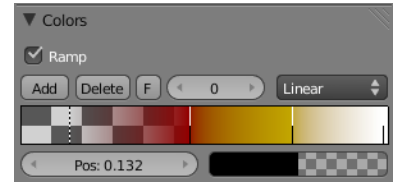
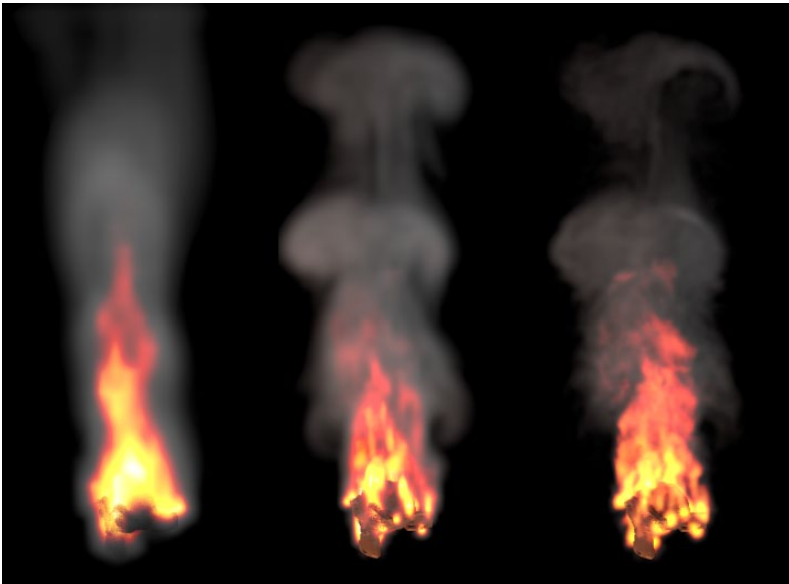


Abb. 8–24

Divisions: 32, 64 und mit High Res Smoke



Sie können jetzt einzelne Bilder mit **F12** berechnen und sollten schon einen guten Eindruck von dem Feuer bekommen. Auch eine Animation sollte sich schnell berechnen lassen. Volumenmaterialien sind übrigens nur mit aktiviertem Ray Tracing zu berechnen. Sind Sie mit der Animation so weit zufrieden, kann die Auflösung der Domain erhöht werden, höchstwahrscheinlich werden Sie dann auch die *Dissolve Time:* erhöhen müssen. Viel kann aber auch schon durch Änderung des Farbverlaufs erreicht werden.

Berechnung und Optimierung

Das Baken (Erstellen des Cache) der Simulation wird durch die höhere Auflösung wesentlich länger dauern. Die Belohnung sind aber detailliertere Flammen und Rauch.

Mit der High Res Smoke-Option wird nur etwas mehr Rechenzeit zum Baken benötigt, dabei werden durch Oversampling mehr Details in den Rauch eingerechnet. Der große Vorteil ist, dass die Bewegung des Rauchs nicht durch die High Res Smoke-Option verändert wird und somit gute, aber zu niedrig aufgelöste Simulationen verbessert werden können.

Blenders Smoke-Simulation reagiert auf alle Kraftfelder, die Sie im Physics Context finden. Erstellen Sie einfach ein Empty und weisen Sie ihm ein

Force Fields

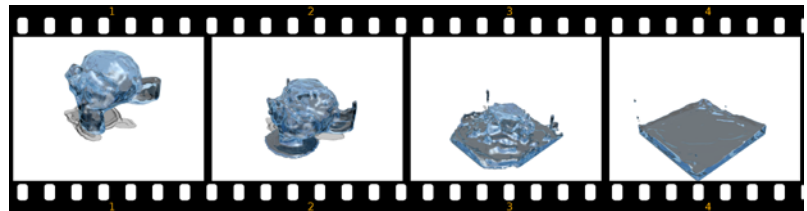
Kraftfeld zu, anschließend müssen Sie die Smoke-Simulation neu backen. Probieren Sie doch einmal das Vortex-Feld für dramatische Effekte, die wie eine Feuerwalze aussehen.

8.6.3 Fluids

Fluid-Simulationen, also die Simulation von Flüssigkeiten, waren wohl eine der ersten physikalischen Simulationen in Blender. Ursprünglich im Rahmen eines Google Summer of Code entwickelt, gestaltete sich die Weiterentwicklung danach etwas schleppend und so funktioniert die Simulation heute noch so gut wie damals. Allerdings ist sie nicht so gut wie Cloth- oder Smoke-Simulationen in den Arbeitsfluss von Blender integriert und reagiert z. B. nicht auf Kraftfelder. Die existierenden Bücher und Online-Tutorials (auch meine DVD [DVD] enthält ein Fluid-Tutorial) sind noch gut mit der Blender-Version 2.5x zu verwenden, daher folgt hier nur eine kurze Einführung.

Über die Fluid-Simulation hinaus gibt es mit den Fluid-Partikeln (siehe Abschnitt 8.5.3) einen flexiblen und einfacher zu handhabenden Konkurrenten in den Startlöchern.

Abb. 8–25
Suzanne total (über-)flüssig



Übung!



Auch bei Fluid-Simulationen brauchen wir eine Domain und mindestens ein Objekt, das die Flüssigkeit im Ausgangszustand definiert. Setzen Sie also Blender zurück und skalieren Sie den Default-Würfel etwas hoch. Setzen Sie dann in die Mitte des Würfels ein Objekt, das Sie »verflüssigen« wollen. Ja, auch hier musste bei meinem Beispiel Suzanne ran (Fluid/Suzanne_Ueberfluessig.blend).

Domain

Selektieren Sie den Würfel, wechseln Sie in den Physics Context und schalten Sie Fluid an. Im Fluid-Panel wählen Sie nun Domain: Daraufhin erscheinen die Parameter und Panels der Domain. Wichtig für ein realistisches Ergebnis sind die Parameter bei Time: und Real World Size: im Domain World-Panel. Time: (in Sekunden) sollte mit der Animationszeit korrelieren, es sei denn, Sie wollen Zeitlupensimulationen machen. Real World Size: muss an die Größe des gewünschten Vorgangs angepasst sein, verhält sich doch ein 5 mm großer Tropfen physikalisch völlig anders als der Schwall eines zehn Liter fassenden Eimers.

