Institut für Visualisierung und Interaktive Systeme
Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Studienarbeit Nr. 2405

"Malprogramm" für Strömungsvisualisierung

Sebastian Konle

Studiengang:	Informatik
Prüfer:	Prof. Dr. Daniel Weiskopf
Betreuer:	Dipl.-Inf. Marcel Hlawatsch
begonnen am:	26. November 2012
beendet am:	28. Mai 2013
CR-Klassifikation:	I.3.3, I.3.4, I.3.8

Kurzfassung

Ziel der Arbeit war eine Art "Malprogramm"für Strömungsvisualisierung zu entwickeln, um verschiedene Strömungsvisualisierungs-Methoden manuell zu kombinieren.

Der Grund Strömungsbilder zu kombinieren ist zum einen der, dass durch beispielsweise zusammenführen von Strömungsbilder, die die Strömung zu verschiedenen Zeitpunkten beschreiben, der Strömungsunterschied zwischen den Zeitpunkten verdeutlicht werden kann. Zum andren können verschiedene Visualisierungsmethoden unterschiedliche Aspekte der Strömung genauer beschreiben. So kann durch das Kombinieren der Techniken der Informationsgehalt eines Bildes erhöht werden.

Jedoch kann es auch zu einem gegenteiligen Ergebnis führen, da die Betrachter entweder visuell überfordert werden oder Informationen durch Überdeckung verloren gehen.

Darum wird in dieser Arbeit ein speziell für dieses Problem entwickeltes Malprogramm zur Strömungsvisualisierung vorgestellt, das unter anderem die Folgenden Funktionen unterstützt:

- eine Ebenenhierarchie um Bilder zu kombinieren
- Malwerkzeuge und Masken um Informationen über Strömungseigenschaften von einem in das andere Bild überführen zu können.
- die Möglichkeit Strömungsbilder aus einem 2D zeitabhängigem Strömungsdatensatz zu erzeugen. Diese können nachträglich nach der Kombination der unterschiedlichen Strömungsbilder verändert werden, um beispielsweise die Strömung zu einem anderen Zeitpunkt zu beschreiben.

Die Bedienoberfläche des Programms wurde mit Qt implementiert und die Grafikausgabe mit OpenGL. Die verwendete Programmiersprache ist C++.

Zusätzlich wird auf Implementierungsdetails eingegangen, die bei einer Strömungsexpertenbasierten Evaluation gesammelten Informationen präsentiert und einige Resultatbilder dargestellt.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Strömungen können durch verschiedene Methoden sichtbar gemacht werden. Beispielsweise durch Verwendung von Farben oder Glyphen. Abbildung 1.1 (links) zeigt eine Strömung, deren Stärke mittels Farben dargestellt wurde. Der Vorteil dieses Verfahrens ist, dass eine hohe Informationsdichte erreicht wurde, da durch jeden Pixel eine andere Strömungsstärke dargestellt werden kann. Jedoch ist das Farbspektrum das zur Verfügung steht um die Strömungsgsstärke zu beschreiben begrenzt. Somit kann bei Strömungen mit zu großer Differenz an unterschiedlichen Stärken nicht zwischen allen Stärken unterschieden werden. Außerdem enthält der in Abbildung 1.1 (links) dargestellte Farbverlauf keine Information über die Richtung der Strömung.

Verwendet man wie in Abbildung 1.1 (rechts) hingegen Pfeile zur Beschreibung der Strömung, kann die Strömungsrichtung durch die Richtung der Pfeile und die Stärke durch deren Länge dargestellt werden. Durch diese Methode ist jedoch die Informationsdichte im Vergleich zur Verwendung von Farben geringer, da die Pfeile entweder zu klein werden oder sich gegenseitig überdecken können.

Jedoch gibt es die Möglichkeit verschiedene Visualisierungsmethoden zu kombinieren, um die Vorteile dieser unterschiedlichen Verfahren nutzen zu können und somit den Informationsgehalt des Resultatbildes zu erhöhen. Dies muss teilweise manuell durchgeführt werde, da es leicht zu einer visuellen Überladung kommen kann und ein Überdeckungsproblem besteht. Dies wird in Abbildung 1.2 verdeutlicht.

Darum beschäftigt sich diese Arbeit mit der Entwicklung eines Malprogrammes für die Strömungsvisualisierung. Hierbei sollen typische Werkzeuge aus Malprogrammen sowie eine Ebenenhierarchie zur Verfügung gestellt werden. Zudem wurden einige Strömungsvisualisierungstechniken implementiert, wodurch im Malprogramm zu jederzeit Strömungsbilder erzeugt, kombiniert und modifiziert werden können.

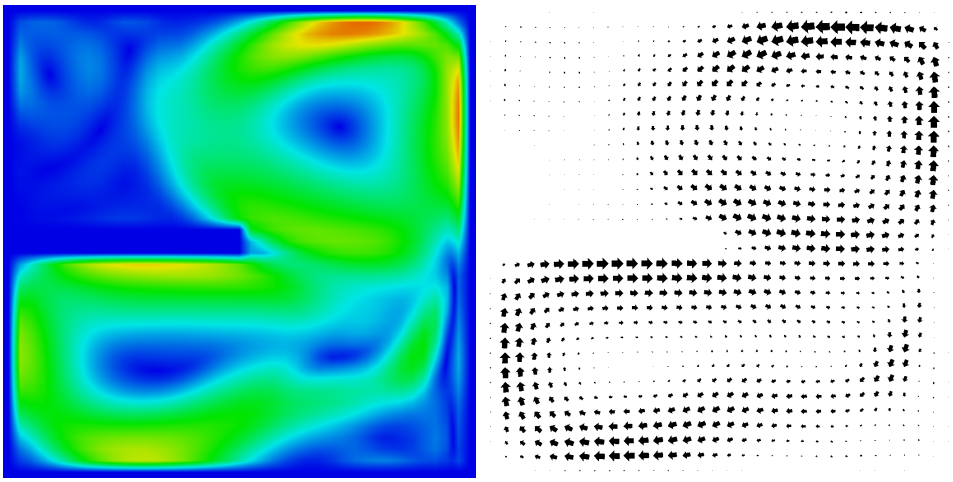


Abbildung 1.1: (links) Die Strömungsgeschwindigkeit wird durch Farben kodiert, jedoch sind keine Informationen über die Richtung der Strömung enthalten. (rechts) Die Strömung wird mit Hilfe von Pfeilen dargestellt. Das so entstandene Bild enthält Informationen über die Richtung und Geschwindigkeit der Strömung. Jedoch kann nur eine begrenzte Menge an Pfeilen dargestellt werden, da sonst die Pfeile zu klein werden.

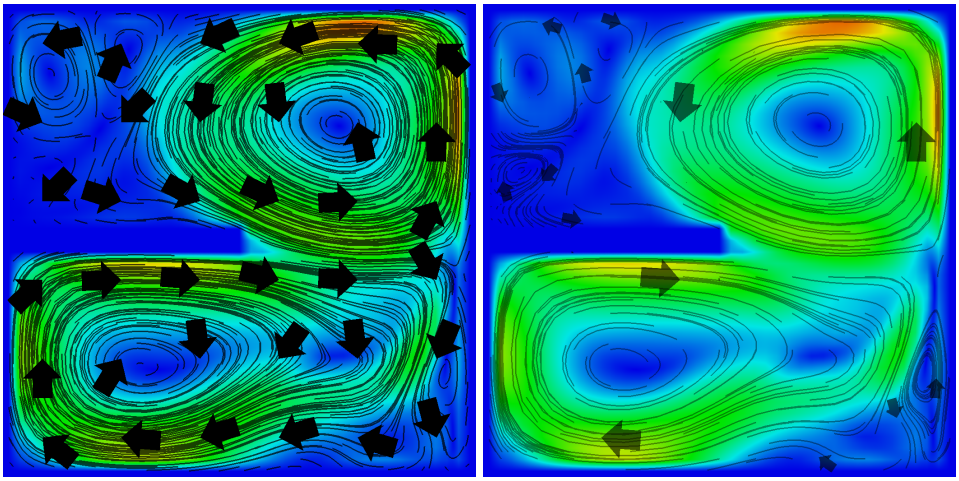


Abbildung 1.2: (links) Die Strömungsbilder wurden ohne Nachbearbeitung kombiniert das Bild wirkt unübersichtlich. (rechts) Das Bild wurde von Hand nachbearbeitet, das Bild wirkt übersichtlicher und das Überdeckungsproblem wurde gelöst.

2 Stand der Forschung

Es gibt verschiedene Arbeiten, die sich damit beschäftigen ein Programm zu entwickeln das es den Benutzern erleichtert, bestimmte Bereiche von Volumen durch direkte Manipulation hervorzuheben beziehungsweise zu untersuchen. Hutchins et al. befassen sich hierbei allgemein in dem Atrikel [ELH85] mit den Vor- und Nachteilen direkter Manipulation.

Im Bereich der Volumen-Darstellung bietet das WYSIWYG (What You See is What Your Get) Volume Visualization System von Guo et al. [HG11] die Möglichkeit mit Echtzeitfeedback bestimmte Bereiche eines Volume-Datensatzes darzustellen.

VolumeShop von Stefan Bruckner und M. Eduard Gröller [SB05] ist ein Programm für direkte Volumen Illustration, durch das dreidimensionale Objekte mit Hilfe von unter anderem transparenten Pfeilen, Rahmen und Ausschnitten näher beschrieben werden können.

Im Artikel Direct Volume Editing von Bürger et al. [KB08] werden neue Werkzeuge und Methoden zur Edition von 3D Volumen vorgestellt.

Die genannten Arbeiten beschäftigen sich zwar mit der selben Problematik Bilder zu erzeugen, die übersichtlich sind und viel Information enthalten. Jedoch sind sie auf 3D Volumendaten ausgerichtet und befassen sich nicht direkt mit der Problematik der Strömungsvisualisierung.

Potentielle Strömungsvisualisierungstechniken, die sich zusätzlich zu den in dieser Arbeit implementierten Methoden eigenen würden, durch ein Malprogramm, kombiniert zu werden sind beispielsweise: Finite-time Layapunov Exponent (FTLE)-Bilder da durch den FTLE von Kasten et al. die Separation von Partikeln in einer Strömung dargestellt werden kann [JK09]. Die Technik Line Integral Convolution (LIC) von Brian Cabral und Leith Leedom , sie verwendet Filterungstechniken um Texturinformationen entlang eines Vektors zu mischen [CL93]. Die Unsteady Flow LIC (UFLIC) Methode von Han-Wei Shen und David L. Kao der die LIC Methode zugrunde liegt aber auf unstetige Strömungsfelder angewendet werden kann. [HWS98] Und die Technik Image Based Flow Visualization (IBFV) von Jarke J. van Wijk, deren Herangehensweise ist, die Texturkoordinaten anhand der Strömungsrichtung und Stärke zu verschieben. [Wij02]

3 Komponenten des Malprogramms

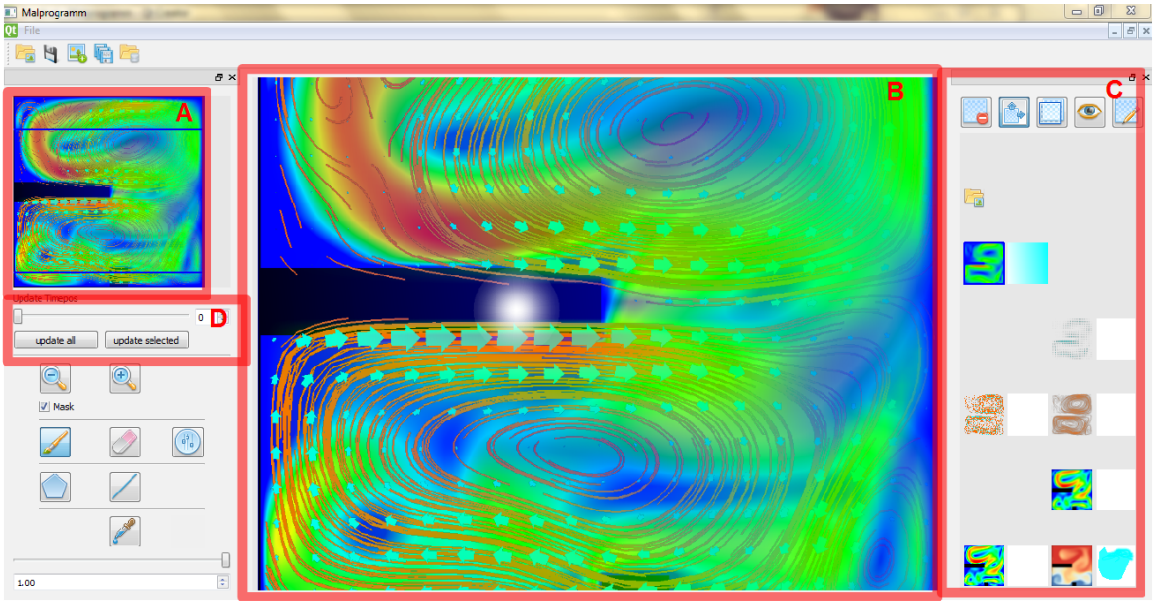


Abbildung 3.1: Dargestellt wird das entstandene Zeichenprogramm. Das sichtbare Feld im Arbeitsbereich (B) wird im Navigationsfenster (A) durch einen blauen Rahmen dargestellt. Durch das Ebenen-Management (C) können die Ebenen und Masken selektiert werden, um sie dann zu modifizieren. Durch die Zeitleiste (D) kännen Strömungsbilder, die mit den in dem Programm integrierten Methoden erzeugt wurden, für einen beliebigen Zeitpunkt neu erstellt werden.(Quelle der Icons: [Hos13])

3.1 Navigation und Überblick

Das Programm besitzt ein Navigationsfenster, das einen Überblick über das Gesamtbild bietet. Siehe Abbildung 3.1 (A) Es zeigt anhand eines blauen Rahmens welcher Bereich momentan im Arbeitsfenster sichtbar ist. Der Arbeitsbereich kann entweder durch Klicken auf eine andere Stelle im Navigationsfenster oder im Arbeitsfeld durch Verschieben der Maus versetzt werden. Das Programm bietet auch eine Zoom-Funktion durch welche der Arbeitsbereich verkleinert oder vergrößert werden kann. (Siehe Abbildung 3.1). Durch die

Zeit-Aktualisierungsleiste (D) können die Zeitinformationen der im Programm erzeugten Strömungsbilder aktualisiert werden, wodurch diese dann neu erstellt werden. Mit Hilfe des Ebenen-Management (C) können Ebenen und Masken selektiert werden, um sie dann zu modifizieren. Die Ebenen werden durch das Miniaturbild der enthaltenen Bildinformationen dargestellt und die Masken durch das Miniaturbild der in der Maske durchgeführten Zeichenoperationen. In Kapitel 3.2 wird näher auf das Ebenen Management eingegangen.

3.1.1 Erzeugen eines Stromlinienbildes

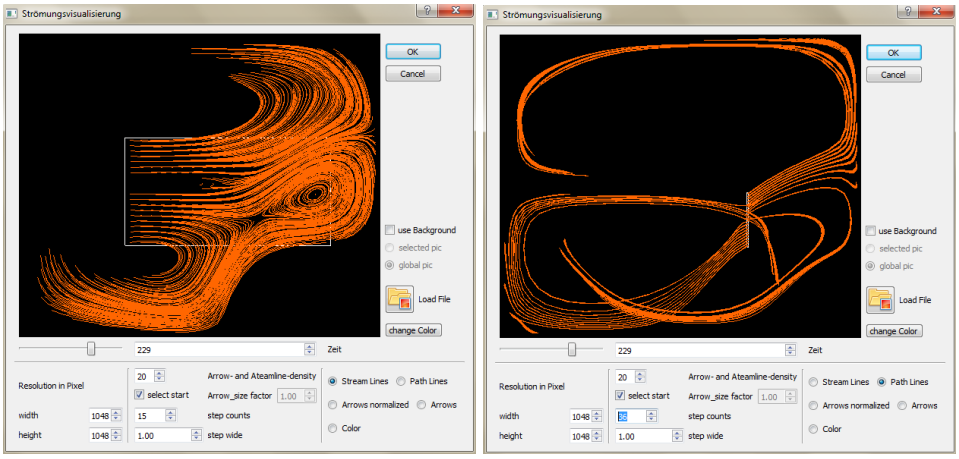


Abbildung 3.2: (links)Stromlinien für den 229ten Zeitschritt werden dargestellt. Durch den weißen Rahmen wird der selektierte Bereich sichtbar gemacht, in dem 20 mal 20 masselose Partikel bei konstanter Zeit für 15 Zeitschritte verfolgt werden, um die Stromlinien zu erzeugen. Mit Hilfe der Zeitleiste (A) kann der gewünschte Zeitpunkt selektiert werden. (rechts) Die zeitabhängige Bewegung der masselosen Partikel des ausgewählten Partikelfeldes wird durch Pfadlinien dargestellt. Die Anzahl der Zeitschritte, die die Partikel verfolgt werden, kann eingestellt werden, sowie die Schrittweite des Runge-Kutta-Verfahrens.

Das Programm bietet die Möglichkeit eine zweidimensionale zeitabhängige Strömung zu laden und Strömungsbilder zu erstellen. Bereits implementierte Strömungsvisualisierungsverfahren sind: Pfeil-Glyphenbild, Farbverlauf, Stromlinien und Pfadlinien.

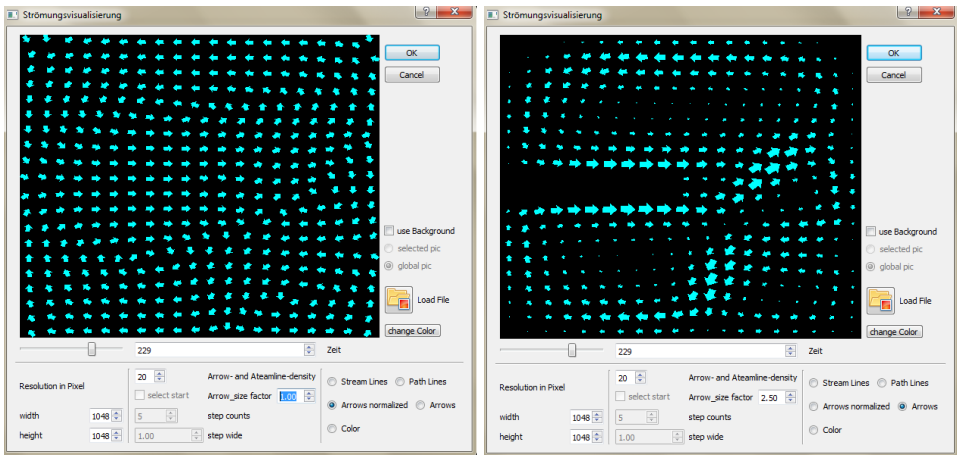


Abbildung 3.3: In den Bildern wird die Strömung durch ein 20 mal 20 großes Feld aus Pfeilen dargestellt. (links) Die Partikelrichtung wird anhand von Pfeilen dargestellt. (rechts) Durch die Pfeilrichtung wird die Bewegungsrichtung der Partikel und durch die Größe der Pfeile die Bewegungsgeschwindigkeit dargestellt.

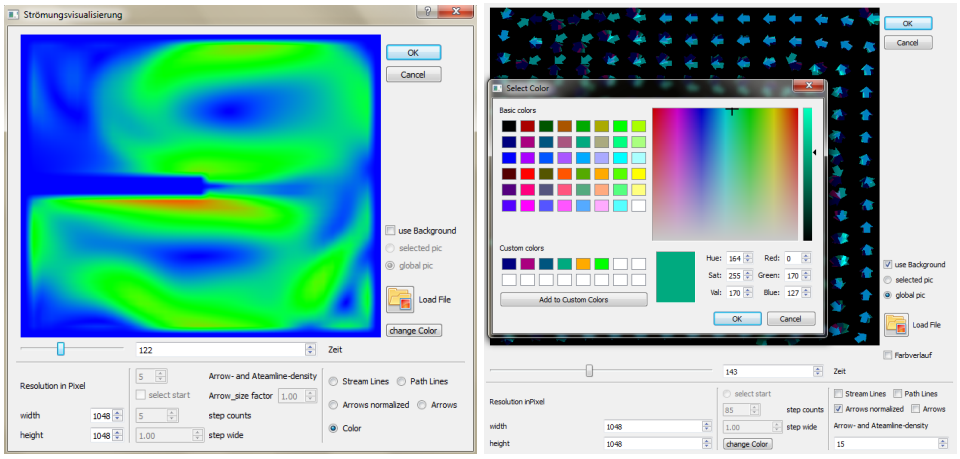


Abbildung 3.4: (links) Die Strömungsgeschwindigkeit wurde durch Farben dargestellt (rechts) Die Farbe der erzeugten Linien und Pfeilen kann selektiert werden.

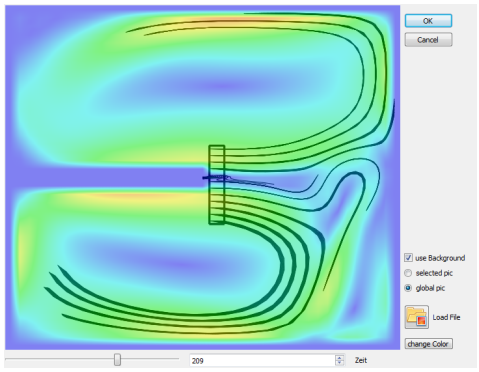


Abbildung 3.5: Im Strömungs-erzeugungsfenster werden Stromlinien erzeugt. Um diese für das Gesamtbild besser einstellen zu können, kann im Hintergrund ein beliebiges Bild dargestellt werden.

Beim Pfeil-Glyphenbild können die erzeugten Pfeile zusätzlich zur Richtung der Strömung, die Strömungsstärke darstellen. Außerdem kann die Größe der Pfeile mit Hilfe eines Faktors vergrößert oder verkleinert werden, da die Pfeilgrößen standardmäßig mit Hilfe des längsten Vektors im gesamten Strömungsfeld normiert wurden, damit sich die Pfeile nicht gegenseitig überlappen. (Siehe Bild 3.3)

Durch das Stromlinienbild kann der Strömungsverlauf dargestellt werden. (Siehe Abbildung 3.2 (links)) Außerdem kann der Startbereich, von dem aus die Stromlinien erzeugt werden, selektiert werden. Dabei wird bei der Berechnung des Verlaufes da Runge-Kutta-Verfahren 4ter-Ordnung verwendet.

Zur Erzeugung der Pfadlinien wird auch das Runge-Kutta-Verfahren verwendet und es kann auch ein Startfeld selektiert werden. Siehe Abbildung 3.2 (rechts) Durch die Pfadlinien kann die Bewegung der gewählten masselosen Partikel zeitabhängig dargestellt werden.

Bei der Erzeugung der verschiedenen Bilder können Einstellungen wie die Auflösung des Resultat-Bildes, oder Farbe und Dichte der Pfeile oder Linien vorgenommen werden. Zudem kann die Schrittzahl der Strom- und Pfadlinien oder die Schrittweite des Runge-Kutta-Verfahrens, verändert werden. (Siehe die Abbildungen 3.4 3.2 3.3)

Um das Erzeugen neuer Strömungs-Bilder zu einem bestimmten Zeitpunkt zu erleichtern, kann ein beliebiges Bild als Hintergrundbild für das Kontrollfenster bei der Erzeugung des Stömungsbildes eingestellt werden. (Siehe Abbildung 3.5) bei der Erzeugung von Stromlinine wurde das Gesamtbild als Hintergrundbild dargestellt.

3.1.2 Laden eines externen Strömungsbildes

Zusätzlich zum Erzeugen neuer Strömungsbilder können auch externe Bilder geladen und verarbeitet werden. So können auch Bilder von nicht im Programm implementierten Strö-

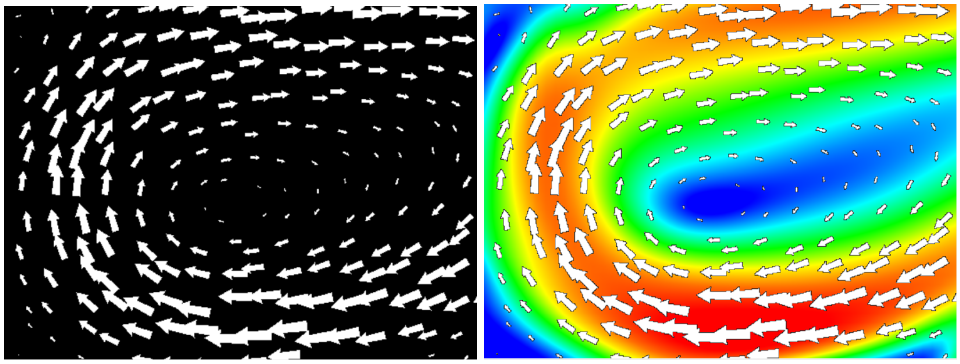


Abbildung 3.6: Ein Bild kann ohne Transparenz im Malprogramm geladenen werden. Durch das Picking-Tool wurde die schwarze Farbe des linken Bildes transparent gemacht, wodurch das darunter liegende Bild zum Vorschein tritt. Das Ergebnisbild ist auf der rechten Seite zu sehen.

mungsvisualisierungsmethoden mit anderen Bildern kombiniert werden. Dabei kann eine Farbe selektiert und unsichtbar gemacht werden. Das macht beispielsweise bei Glyphenbildern Sinn, wenn die geladenen Texturen keinen Transparenz enthalten und man nur die Pfeile sichtbar machen möchte. (Siehe Abbildung 3.6)

3.2 Ebenen Management

Ebenen werden verwendet, um Bilder beliebig übereinander zu zeichnen. Das Ebenen Management ist ein zentraler Bereich des Malprogramms. Mit ihm können Ebenen selektiert, transparent, gelöscht oder unsichtbar gemacht und durch die Drag and Drop Methode verschoben werden. Wurde eine Ebene unsichtbar gemacht, wird dies durch ein rotes Kreuz im Miniaturbild dargestellt.

Um zu erkennen, welche Ebene zu welchem Bild gehört, wird die Ebene durch ein Miniaturbild des dazugehörigen Bildes repräsentiert. Dieses Miniaturbild entspricht immer dem aktuellen Bild und wird daher bei jeder Änderung aktualisiert.

Die Ebenen können auch hierarchisch angeordnet werden, damit Änderungen auf einer übergeordneten Ebene an deren Kinder (Ebenen, auf die von der momentanen Ebene verwiesen wird) weitergegeben werden oder auch Ebenen zusammen mit all ihren Kindern verschoben werden können.(Siehe Abbildung 3.7)

Wurde ein Bild im Malprogramm mit integrierten Techniken erzeugt, kann dieses Bild jederzeit nachträglich modifiziert und neu erstellt werden. Möchte man beispielsweise die Anzahl der erzeugten Pfeile im Bild ändern, kann man das erzeugte Pfeilbild selektieren und nachträglich modifizieren.

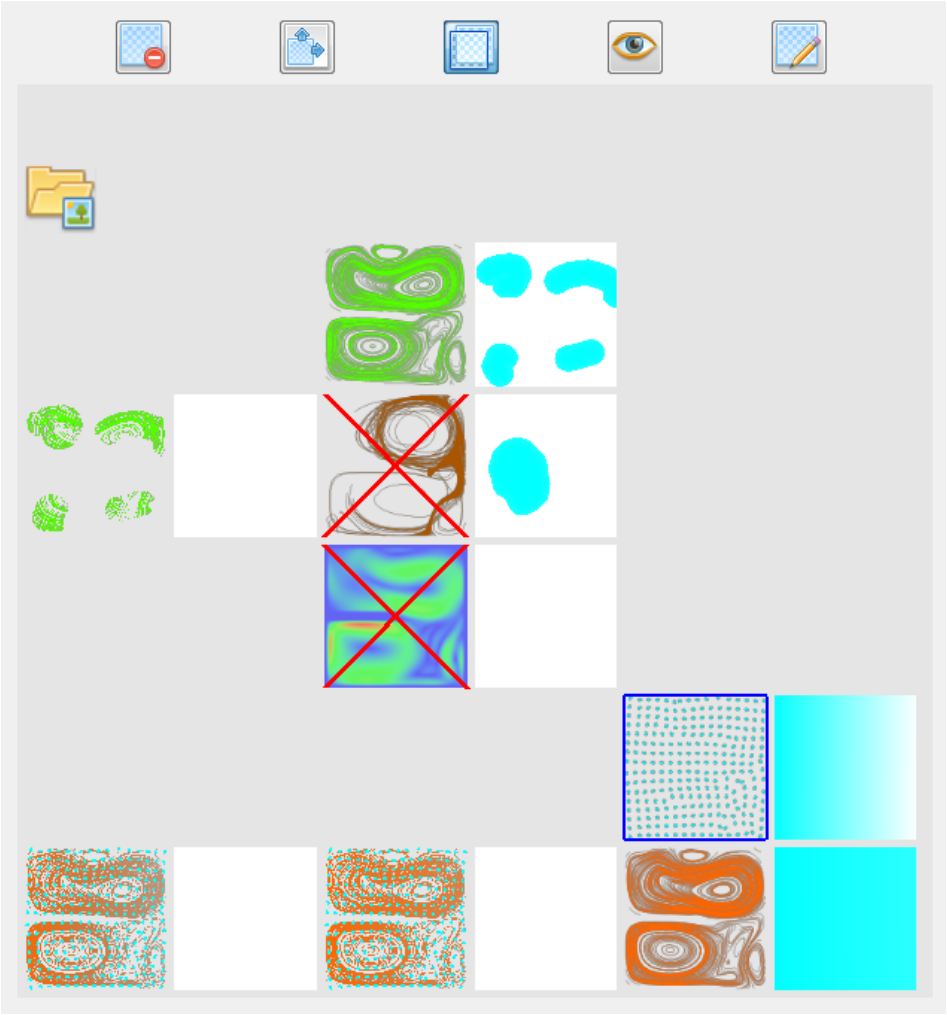
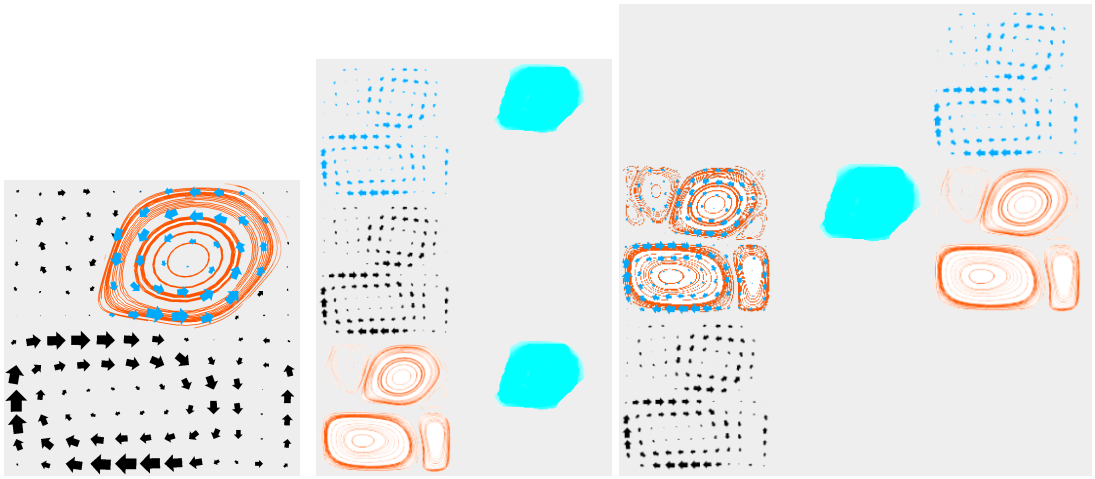


Abbildung 3.7: Im Ebenenfenster werden die Ebenen und Masken dargestellt. Dort können Ebenen gelöscht, verschoben, ausgeblendet und bearbeitet werden. Masken werden rechts neben den zu ihnen gehörenden Ebenen dargestellt. Sie können in dem Ebenenfenster gelöscht, vertauscht und zusammengefasst werden. Ausgeblendete Ebenen werden durch ein rotes Kreuz markiert. Die Ebenen-Hierarchie wird im Ebenenfenster von links nach rechts dargestellt. Auf der rechten Seite zu den Ebenen befinden sich immer die nächsten Kinder wobei diese sich immer mindestens auf der selben Höhe zu ihrem Vater befinden.

3.2.1 Masken Management

Masken werden verwendet, um ohne die Originaltextur zu verändern, Änderungen am Bild durchzuführen. Dies geschieht indem z.B. beim Zeichnen mit dem Pinselwerkzeug nur in die dazugehörige Maske gezeichnet wird und diese Informationen dann zusammen mit der Originaltextur verarbeitet werden um das Ergebnisbild zu erzeugen.



(a) (b) (c)

Abbildung 3.8: Das Ergebnisbild (a) Ergibt sich entweder durch (b) die Kombination von 3 Bildern und zwei identischen Masken oder durch (c) indem eine Hierarchie verwendet wird, wodurch nur eine Maske benötigt wird.

Die Masken werden auch in der Ebenen-Hierarchie dargestellt. Wie die Ebenen werden die Masken auch durch Miniaturbilder der in der Maske enthaltenen Informationen repräsentiert und aktualisiert. Das Programm bietet die Möglichkeit die Masken verschiedener Ebenen zu tauschen, löschen und zu kombinieren. Wobei das Kombinieren und Tauschen durch Drag and Drop realisiert wurde. (Siehe Abbildung 3.7) Der Aufwand, Masken zu kopieren um bestimmte Bereiche aufeinander anzupassen, wird mit Hilfe der Ebenen-Hierarchie stark vereinfacht, da dadurch eine Maske auf mehrere Ebenen angewendet werden kann und so bei Änderungen der gemeinsamen Masken keine Maskeninformation mehr kopiert werden muss. (Siehe Abbildung 3.8)

3.3 Werkzeuge

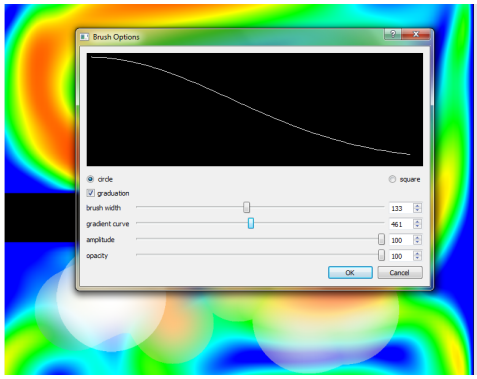


Abbildung 3.9: Im Pinseloptionsfenster können die Breite und Deckkraft des Pinsels und die Deckkraft zum Pinselrand hin eingestellt werden. Das Abfallen der Deckkraft des Pinsel entspricht einer Gaußkurve und wird im Optionsfenster angezeigt. Außerdem kann die Amplitude der Gaußkurve eingestellt werden.

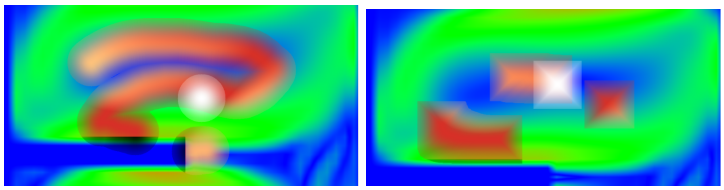


Abbildung 3.10: Je nach Pinseleinstellung wir der aktive Pinsel an der Position des Cursors angezeigt

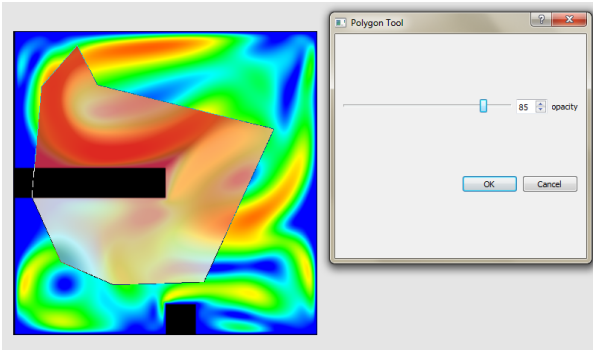
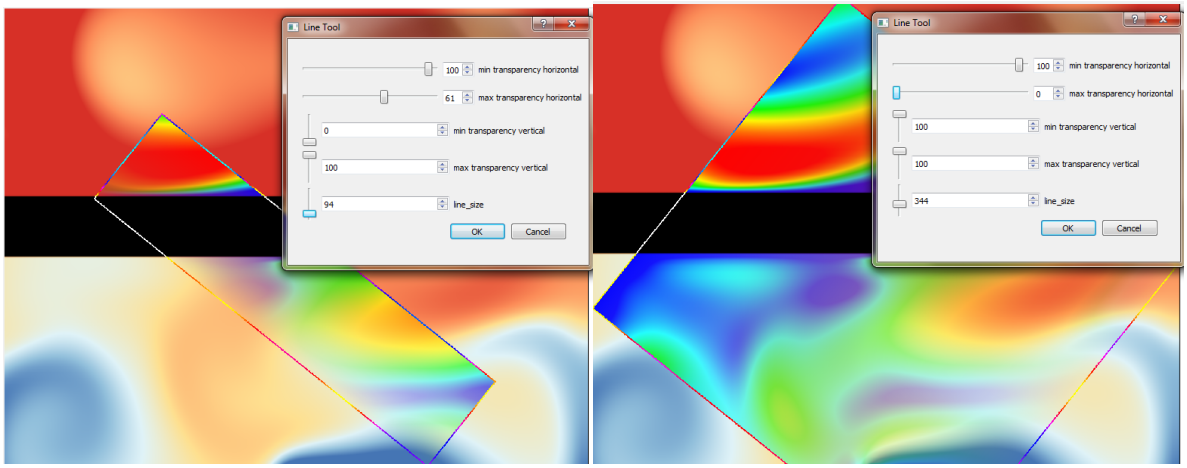


Abbildung 3.11: Im Polygonerzeugungsmodus kann ein beliebiges Polygon in das Bild gezeichnet werden. Dabei kann die Deckkraft des Polygons verändert werden, bis das Polygon dann endgültig in das Bild gezeichnet wird. Durch den Rahmen um das Polygon wird dargestellt dass das Polygon noch nicht endgültig in die Maske gezeichnet wurde.



(a) **(b)**

Abbildung 3.12: Im Linien-Zeichenmodus kann eine beliebige Linie erzeugt werden. Dabei können Start- und Endpostion per Mausklick ausgewählt werden. Die Deckkraft der Linie kann horizontal und vertikal zur Linie verändert werden. Nachdem die gewünschte Linie bestätigt wurde wird der Rahmen entfernt und die Linie fest in die Maske gezeichnet.

Das Malprogramm bietet die folgenden Werkzeuge um die zu den Ebenen gehörenden Masken zu manipulieren: Pinsel-, Radier-, Linien- und Polygon-Werkzeug. Dabei wird im Ebenen Management Fenster die gewünschte Ebene selektiert, um zu definieren welche Maske verändert werden soll.

Zu dem Pinsel- und Radier-Werkzeug kann die Breite und die Form (Eckig oder Rund) eingestellt werden. Zusätzlich kann beim Pinsel die Deckkraft graduell abfallen oder im Gesamten eingestellt werden. (Siehe Abbildung 3.9) Als Rückmeldung, dass Radierer oder Pinsel aktiv sind, wird ein Bild des aktuelle Pinsels bzw. Radierers an der Curserposition angezeigt. (Siehe Abbildung 3.10)

Das Polygonwerkzeug bietet die Möglichkeit beliebige Punkte aus dem Bild durch Mausselektion auszuwählen und dann die gewünschte Deckkraft zu bestimmen. Als Rückmeldung für die Einstellung wird zur Demonstration ein Polygon an die selektierte Position gezeichnet und erst dann fest in die Maske geschrieben, wenn die Einstellung akzeptiert wurde. (Siehe Abbildung 3.11)

Dasselbe gilt für das Linientool (siehe Abbildung 3.12) wobei hier durch Selektion eines Anfangs- und Endpunktes eine Linie gezeichnet wird, bei der zusätzlich der vertikale und der horizontale Gradient für die Deckkraft eingestellt werden kann.

3.4 Speichern und Laden

Im Programm kann jederzeit das aktuelle Projekt mit seinen Einstellungen gespeichert und neu geladen werden. Dabei werden die verschiedenen Ebenen- und Masken-Bilder als PNG gespeichert. Damit können sie direkt aus dem Speicherverzeichnis genommen und weiterverwendet werden. Zudem kann das finale Bild separat gespeichert werden.

4 Implementierungsdetails

Im Folgenden werden wesentliche Punkte der Implementierung beschrieben, die die Funktionsweise des Malprogramms beschreiben. Im Besonderen wird bei der Implementierung auf ressourcenschonende und rasche Abwicklung der einzelnen Schritte der Datenverarbeitung geachtet. Wo immer möglich werden dabei Kopiervorgänge vermieden und stattdessen Verweise eingesetzt.

4.1 Erzeugen eines Bildobjekts

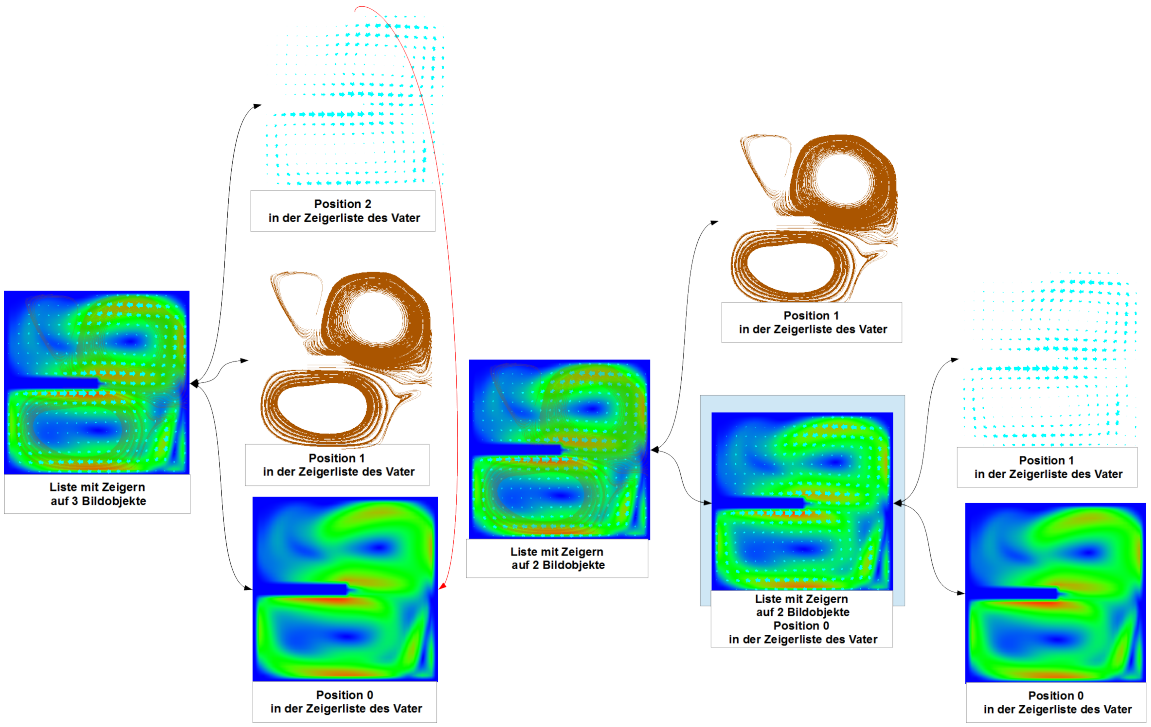
Zu jedem Bild wird ein Bildobjekt erzeugt, das die IDs der auf die Grafikkarte geladenen Textur, Maske und die jeweiligen Frame Buffer Objekte enthält. Zusätzlich befindet sich eine Liste, in die Zeiger auf Bildobjekte eingefügt werden können, in dem Objekt. Diese Liste spielt eine zentrale Rolle im Ebenen-Management da dadurch die Ebenen-Hierarchie aufgebaut wird.

4.2 Ebenen Management

Das Ebenen Management greift auf ein globales Bildobjekt zu, von dem aus (durch die enthaltene Liste) auf alle Bildobjekte der Hierarchiestufe 1 verwiesen wird. Um die Ebenen darzustellen, werden diese Bildobjekte nacheinander durchlaufen und ihre Texturen gezeichnet. Zudem wird im Render Modus: GL_SELECT von OpenGL die Listenposition des Objekts und ein Zeiger auf die Liste als Kennung gespeichert. So kann durch Mausselektion ein Objekt jederzeit identifiziert werden. Um zwei Bildobjekte zu kombinieren, wird ein neues Bildobjekt erzeugt und in dessen Zeigerliste, zwei auf die zu kombinierenden Objekte verweisende Zeiger eingefügt. (Die ehemaligen Zeiger auf die Bildobjekte werden dabei gelöscht.) Danach wird das Frame Buffer Objekt des neuen Bild-Objektes aktualisiert. (Siehe Abbildung 4.1)

4.3 Aktualisierung eines Bildobjekts

Um ein Bildobjekt zu aktualisieren, wird ein Frame Buffer Objekt (FBO) verwendet. Dadurch kann eine Textur direkt auf der Grafikkarte manipuliert werden. Dies hat den Vorteil, dass die Texturen die für das Bild im Bildobjekt kombiniert werden, direkt auf der GPU bleiben und



(a) **(b)**

Abbildung 4.1: (a) Durch den roten Pfeil wird dargestellt, dass das Bildobjekt mit den Pfeilinformationen mit dem in dem der Farbverlauf dargestellt wird, gruppiert werden sollen. Bei der Selektion werden die Listen und die Postionen der darin enthaltenen Zeigers erkannt. (b) Nachdem die Bildobjekte gruppiert wurden wurde ein neues Bildobjekt erstellt (im hellblauen Kasten). Durch die Zeigerliste dieses Bildobjektes wird auf die gruppierten Bildobjekte verwiesen. Da sich durch diesen Vorgang die Reihenfolge der Bildobjekte verändert haben kann, werden alle Bildobjekte, die von dem neuen Bildobjekt aus auf dem Weg zum Wurzelbildobjekt befinden, aktualisiert.

verarbeitet werden können, um die Ergenis-Textur zu erhalten. Bei der Aktualisierung eines Bildobjekts wird in dessen Zeiger-Liste überprüft, ob das Bildobjekt aus anderen Bildobjekten zusammengesetzt wurde. Ist dies der Fall, wird das FBO aktiviert mit welchem die Bildtextur des aktuellen Objektes manipuliert werden kann und die Bildtexturen der Bildobjekte auf die durch die Zeigerliste verwiesen wird, werden gezeichnet. Wurde der Inhalt eines Bildes durch beispielsweise Zeichnen oder Verschieben von Bildobjekten geändert, werden alle von dem geänderten Bildobjekt aus auf dem Weg zur Wurzel liegenden Bildobjekte aktualisiert.

4.4 Masken

Die Maske besteht aus einem FBO und einer Maskentextur, die mit Hilfe des FBOs verändert werden kann. Da jeder Ebene eine Maske zugewiesen wird, kann das Vertauschen der Maske durch einfaches Umhängen von Zeigern durchgeführt werden.

Das Kombinieren von Masken wird mit Hilfe des FBOs auf der GPU durch Zeichnen der Texturinformationen der einen Maske in die Textur der anderen Maske durchgeführt. Da sich die beiden Maskentexturen im Speicher der Grafikkarte befinden, kann das Verbinden der Informationen einfach im Fragmentshader durchgeführt werden. (Siehe Abbildung 4.2)

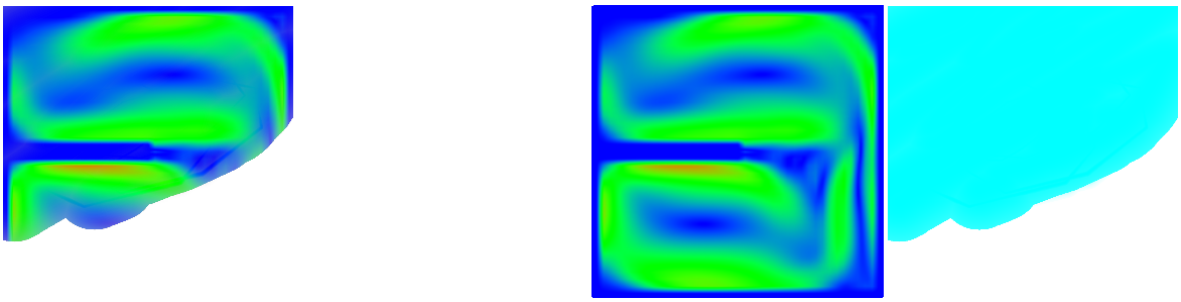


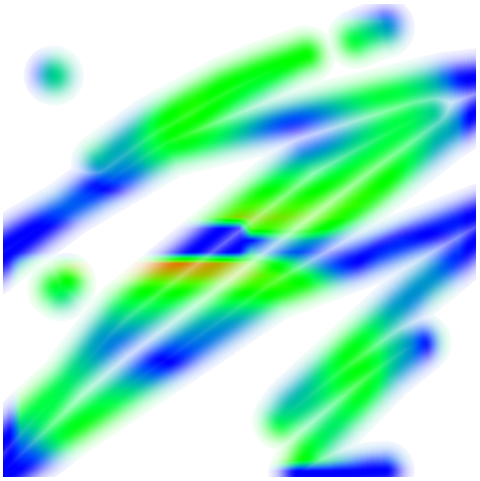
Abbildung 4.2: Auf der linken Seite ist das Ergebnis zusehen, nachdem die Textur und Maske auf der rechten Seite im Fragment Shader kombiniert wurden. Dies wurde durchgeführt, indem die alpha-Werte in der Maske (rechts) türkis dargestellt mit dem alpha-Werten der Bildtextur kombiniert wurden.

4.5 Werkzeuge

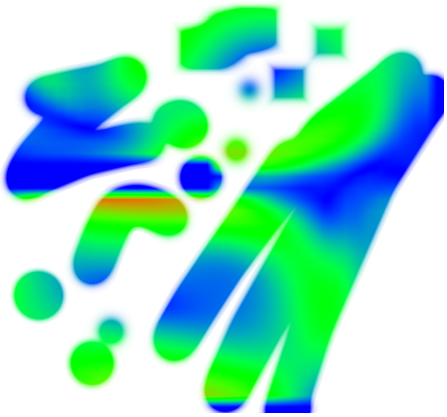
Die im Malprogramm implementierten Zeichenwerkzeuge werden auf ähnliche Weise realisiert. Sie zeichnen direkt in den alpha-Kanal der Maskentextur. Die Informationen der Maskentextur und die der Bildtextur werden dann im Fragment Shader zusammengeführt (dabei wird der in der Maskentextur gespeicherte alpha-Kanal mit dem der Bildtextur zusammengefasst indem sie multipliziert werden). Dies hat den Vorteil, dass die Bildtextur nicht verändert wird und somit angewendete Zeichenoperationen ohne Probleme ausgetauscht oder gelöscht werden können.

4.5.1 Pinsel-,Radier-Werkzeug

Mit Hilfe der selektierten Pinseleigenschaften wird eine Pinseltextur erzeugt, die dann in den Grafikkartenspeicher geladen wird. Diese Textur wird dann mittig zur Mausposition in die Textur der Maske mit Hilfe eines Frame Buffer Object (FBO) gezeichnet.



(a) Ergebnis nach Verwendung des Pinsels, wobei die Blendgleichung auf MAX gestellt wurde also der Maximale Pixel zwischen dem neu gezeichneten und schon vorhandenem Pixel gewählt wird.



(b) Bei der voreingestellte Bledgleichung GL_FUNC_ADD wirkt das Zeichnen in den alpha-Kanal wie Aufgesprüht, da sich die alpha-Werte der Maske und des Pinsels addieren.

Abbildung 4.3

(a) Durch die Verwendung des Pinsels und die Einstellung der Bledgleichung auf GL_FUNC_REVERSE_SUBTRACT wird der alpha-Wert des Pinsels von dem in der Maske enthaltenen apha-Wert subtrahiert.

(b) Ergebnis nach Verwendung des Radierers die Blendgleichung wurde auf MIN gestellt dabei wurden die alpha-Werte, die mit dem Radierer in Berührung mit dem Radierer gekommen sind, auf 0 gesetzt.

Abbildung 4.4

Dies hat den Vorteil, dass durch die Verwendung des FBOs beim Zeichen nur vier Vertex-Koordinaten, um die Position und Größe des Pinsels zu definieren, und vier Textur-Koordinaten, um die Textur aufzuspannen, an die Grafikkarte geschickt werden muss und dort dann die Pinsel-Textur verwendet werden kann um die Informationen in die Maske zu schreiben. Anderenfalls, müssten die Pinsel-Informationen bei jedem Zeichschritt auf die Grafikkarte geladen werden, um in die Masken-Textur zu zeichnen, was einen höheren Zeitaufwand zur Folge hätte.

Ändert sich die Mausposition schnell, wird zwischen den Positionen interpoliert und die Pinseltextur auf die jeweiligen Positionen in der Maske gezeichnet. Dabei wird die Blendgleichung des Frame Buffer Objects so eingestellt, dass das Maximum zwischen dem sich schon im FBO befindlichen alpha-Wertes und des neu gezeichneten alpha-Wertes des Pinsels im FBOs gespeichert wird. Mit Hilfe der Blendgleichung wird erreicht, dass ein realistischer Pinseleffekt entsteht. (Siehe Abbildung 4.3 (links)). Wurde dies nicht gemacht, ist die Verarbeitung der Pixel standardmäßig auf GL_FUNC_ADD eingestellt was bedeutet, dass die Pixelwerte addiert werden bis sie das Maximum erreicht haben. (Siehe Abbildung 4.3 rechts, anstatt des Pinsel-Effekts entsteht dann ein Spray-Effekt.) In Abbildung 4.4 ist zusehen, dass man mit dem Befehl GL_FUNC_REVERSE_SUBSTRACT den Spray-Effekt invertieren kann. Somit könnten alternative Werkzeuge zum Pinsel und Radierer realisiert werden.

Aus dem Pinsel-Werkzeug kann leicht ein Radier-Werkzeug gemacht werden, indem die Pinsel-Textur angepasst wird (alpha-Werte die größer 0 sind werden zu 0 und die die gleich 0 sind auf 1 gesetzt) und das FBO auf Übernahme des Minimalen alpha-Wertes eingestellt wird.

4.5.2 Linien- Polygon-Werkzeug

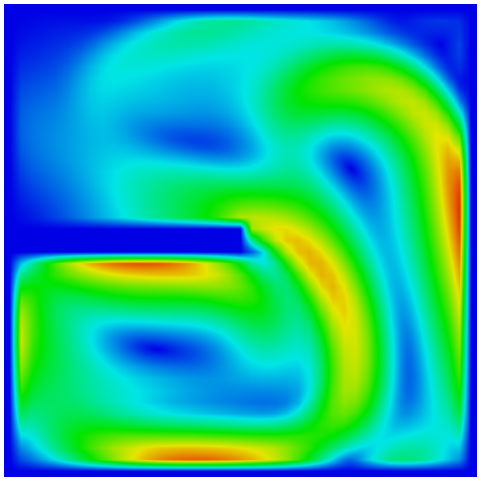
Mit dem Linien-Werkzeug können rechteckige Bereiche erzeugt werden. Diese werden dann in die Maskentextur gezeichnet. Dabei kann im Fragment Shader ein graduelles Abfallen des alpha-Kanals erzeugt werden. Durch vorheriges Kopieren der Maskentextur kann das Zeichnen in die Maske rückgängig gemacht werden und somit eine Vorschau für die erzeugte Linie erstellt werden. Das Polygon-Werkzeug funktioniert auf dieselbe Weise, nur dass eine globale Transparenz verwendet wird und kein graduelles Abfallen.

5 Strömungsvisualisierung

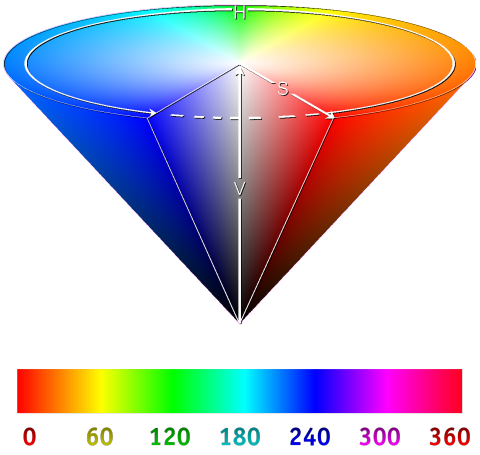
Im folgenden Kapitel werden die im Malprogamm implementierten Strömungsvisualisierungstechniken näher beschrieben. Hierbei werden noch einmal die Eigenschaften der verschiedenen Visualisierungstechniken aufgeführt und dann die Implementierung erklärt.

5.1 Farbverlauf

Mit Hilfe des Farbverlaufbildes (siehe Abbildung 5.1a) wird die Strömungsgeschwindigkeit dargestellt. Dies hat den Vorteil, dass die Informationsdichte sehr hoch ist, da jeder Pixel eine andere Geschwindigkeitsinformation darstellen kann. Jedoch wird die Bewegungsrichtung der Strömung mit dieser Methode nicht dargestellt. Dadurch, dass der Farbbereich der für



(a) Die Strömungsgeschwindigkeit wird mit Hilfe von Farben dargestellt.



(b) Anhand des HSV Farbraumes kann der Farbverlauf bestimmt werden. Der maximal Wert ist rot und der minimal Wert blau. [1]

Abbildung 5.1

[1] https://commons.wikimedia.org/wiki/File:HSV_cone.png
[1] https://commons.wikimedia.org/wiki/File:HueScale.svg

die Darstellung verwendet wird beschränkt ist, ist auch die Menge der unterschiedlichen Geschwindigkeiten, die durch die Farbcodierung dargestellt werden kann, beschränkt.

5.1.1 Implementierung des Farbverlaufs

Der Farbverlauf wird im Fragment Shader erzeugt. Dazu muss vor der Berechnung die maximale Stärke der Strömung, die Position des Strömungsfeldes im Speicher sowie der momentan betrachtete Zeitpunkt an den Fragment Shader übergeben werden.

Danach kann der Shader mit Hilfe der Texturkoordinaten und dem momentanen Zeitpunkt jedem Pixel eine Strömungsstärke zuweisen. Die von der Strömungstextur zugewiesene Stärke wird dann entsprechend der maximalen Stärke auf den Bereich zwischen $0° - 240°$ abgebildet und dann mit Hilfe des HSV Farbspektrums ein Farbwert bestimmt. Die Umrechnung vom HSV-Farbraum in den RGB Farbraum kann mit Hilfe der Formel 5.1b durchgeführt werden. Dabei ist zu bemerken, dass das verwenden des HSV-Farbverlaufes sich zwar anbietet, da es relativ leicht zu implementieren ist, es jedoch keine optimale Lösung darstellt, weil laut David Borland und Russell M. Taylor II [DB07] der Farbverlauf zu Fehlinterpretation führt, da durch den Helligkeitsunterschied der Farben Gradienten vorgetäuscht werden, die nicht existieren.

$$(5.1)\quad m_1 = S \sin H, \qquad m_2 = S \cos H, \qquad i_1 = \frac{V}{\sqrt{3}}$$

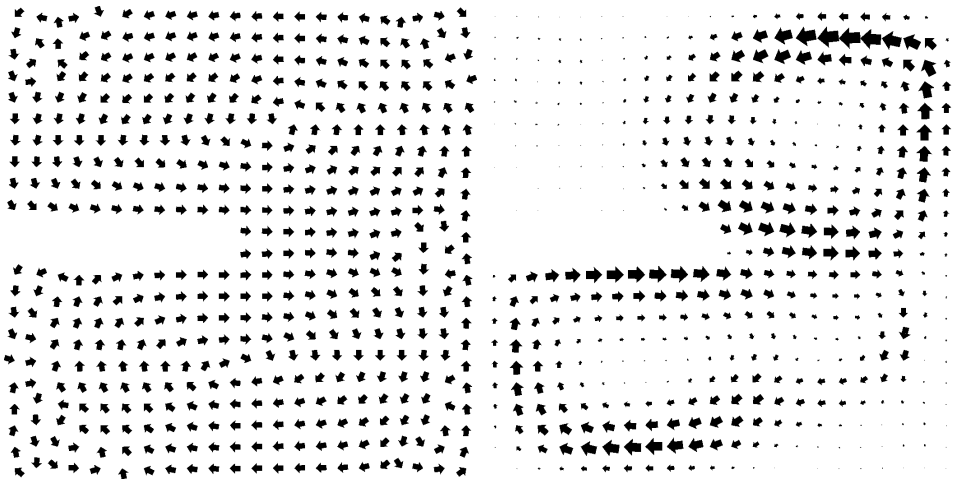
und

$$(5.2)\quad \begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} \frac{2}{\sqrt{6}} & 0 & \frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \end{pmatrix} * \begin{pmatrix} m_1 \\ m_2 \\ i_1 \end{pmatrix}$$

(Quelle: [AN11])

5.2 Strömungsdarstellung durch Pfeile

Verwendet man Pfeile um Strömungen darzustellen, kann man zwei Arten von Strömungsinformationen darstellen. Zum einen kann mit der Pfeilrichtung die Bewegungsrichtung der Strömung darstellen werden (siehe Abbildung 5.2a), zum anderen kann die Stömungsstärke anhand der Pfeilgröße sichtbar gemacht werden, was in Abbildung 5.2b zu sehen ist. Ein großes Problem bei dieser Technik ist jedoch, dass bei der Darstellung zu vieler Pfeile, diese sich entweder überdecken oder sie so klein werden, dass nicht mehr erkannt werden kann in welche Richtung sie zeigen.



(a) Darstellung der Strömung durch Pfeile. Hierbei ist nur die Information über die Strömungsrichtung enthalten.

(b) Dastellung der Strömung durch Pfeile die zusätzlich zur Richtungsinformation auch die Geschwindigkeit durch ihre Länge ausdrücken.

Abbildung 5.2

5.2.1 Implementierung der Strömungsdarstellung durch Pfeile

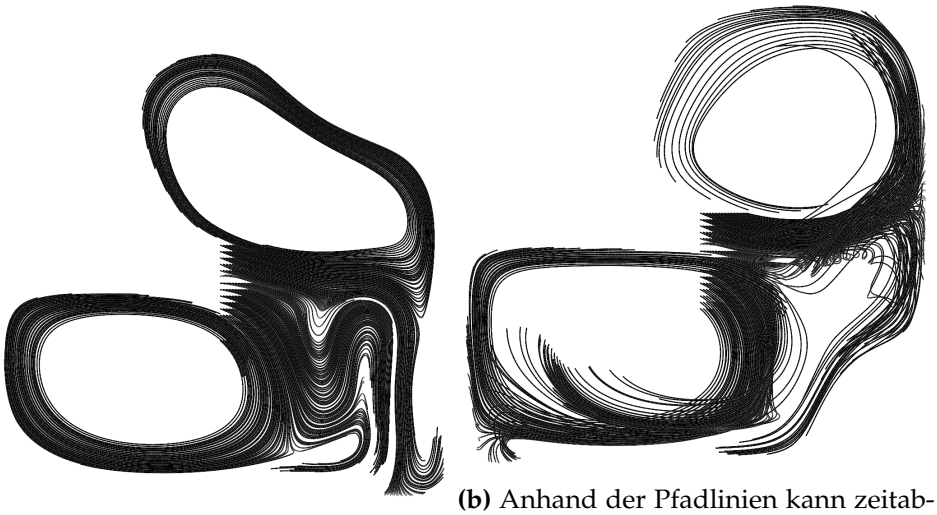
Um die Strömung mit Pfeilen darzustellen, wird ein Gitter mit Vertex Koordinaten, auf die die Pfeile erzeugt werden sollen, auf die Grafikhardware geladen. Dort werden dann im Geometry-Shader für jede Koordinate der dazu passende Vektor aus dem Strömungstextur geladen und der Winkel des Vektors bestimmt. Dieser Winkel entspricht der Strömungsrichtung an dieser Position. Nun kann der Geometry Shader 8 neue Vertex-Koordinaten erzeugen um einen Pfeil zu zeichnen, der in die gewünschte Richtung zeigt. (Dabei werden die Positionen abhängig zur erhaltenen Vertex-Koordinate gesetzt, damit der Pfeil an der richtigen Position erscheint).

Die Länge des Pfeiles wird berechnet, indem die Größe des Feldes durch die Anzahl der Pfeile geteilt wird. (Befindet sich die Spitze des Pfeiles wie in Abbildung 5.2 auf der Position des betrachteten Punktes, muss die berechnete Pfeillänge zusätzlich halbiert werden, damit sich die erzeugten Pfeile nicht überschneiden können.)

Möchte man einen Pfeil erzeugen, durch dessen Größe die aktuelle Strömungsstärke dargestellt wird, wird zusätzlich die maximale Strömungsstärke im Feld benötigt. Mit ihr und der aktuellen Strömungsstärke kann dann die Länge des Pfeiles angepasst werden.

5.3 Pfadlinien

Pfadlinien werden verwendet, um den Weg eines masselosen Partikels in der Strömung darzustellen. Dies geschieht, indem anhand von Linien gezeigt wird, entlang welchem



(a) Durch die Stromlinien wird der Verlauf der Strömung dargestellt

(b) Anhand der Pfadlinien kann zeitabhängig die Bewegungsrichtung eines Partikels dargestellt werden.

Abbildung 5.3

Weg sich Partikel in einem zeitabhängigem Strömungsfeld bewegen würden. Die Richtung des Partikel kann zwar teilweise mit dem Wissen wo sich die Startposition befunden hat interpretiert werden, jedoch ist diese Information bei geschlossenen Linien nicht mehr von Nutzen.

5.3.1 Implementierung der Pfadlinien

Pfadlinien werden im Geometrie Shader erzeugt und die Startpositionen als Vertex-Array auf die Grafikkarte geladen. Dadurch wird die Berechnung erheblich beschleunigt, da somit die unterschiedlichen Stromlinien parallel berechnet werden können. Zu jeder Vertex-Koordinate werden dann mit Hilfe des Runge Kutta-Verfahrens vier Zwischenpunkte aus des Strömungstextur geladen und eine neue Vertexkoordinate erzeugt. (Siehe Formel (5.3)) Danach wird eine Verbindungslinie von der übergebenen Vertex-Koordinate zur neu erzeugten Koordinate gezeichnet. Da der Geometrie-Shader nur eine begrenze Anzahl von Punkten erzeugen kann, muss, wenn der Shader sein Maximum erreicht hat, der letzte Punkt zwischengespeichert werden. Mit diesen zwischengespeicherten Koordinaten wird der Aufruf zum Zeichnen von neuem gestartet. Dabei muss beachtet werden, dass der Texturspeicher des Frame Buffer Objekts in das gezeichnet wird nicht gelöscht wird, da sonst die schon gezeichneten Linien verloren gehen.

(5.3)

$$
\begin{aligned}
k_1 &= h * f(v_{n.x}, v_{n.y}, v_{n.z}),\\
k_2 &= f(v_n.x + \frac{1}{2} * hk_1.x, v_n.y + \frac{1}{2} * hk_1.y, tn + \frac{1}{2} * hk_1.z),\\
k_3 &= f(v_n.x + \frac{1}{2} * hk_2.x, v_n.y + \frac{1}{2} * hk_2.y, t_n + \frac{1}{2} * hk_2.z),\\
k_4 &= f(v_n.x + hk_3.x, v_n.y + hk_3.y, t_n + \frac{1}{2} * hk_1.z, t_n + \frac{1}{2} * hk_3.z)\\
v_{n+1} &= v_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned}
$$

In dieser Formel wird vom Punkt (x_n, y_n, z_n) der nächste Punkt $(x_n + h, y_n + h, z_n + h)$ berechnet. $k_{1..4}$ stellen die vier Zwischenstufen dar, die für die Rechnung mit einbezogen werden. Hierbei enthält z_n die Zeitinformation.

5.4 Stromlinien

Stromlinien werden verwendet, um den momentanen Strömungsverlauf darzustellen. Wenn sich die Strömung über die Zeit hinweg nicht verändert, haben sie den gleichen Verlauf wie Pfadlinien.

Darum ist die Implementierung im Prinzip identisch zur Implementierung der Pfadlinien. Der einzige Unterschied ist, dass der betrachtete Zeitpunkt der Strömung konstant ist und deswegen bei der Durchführung des Runge-Kutta-Verfahrens konstant bleibt.

6 Expertenbasierte Evaluation

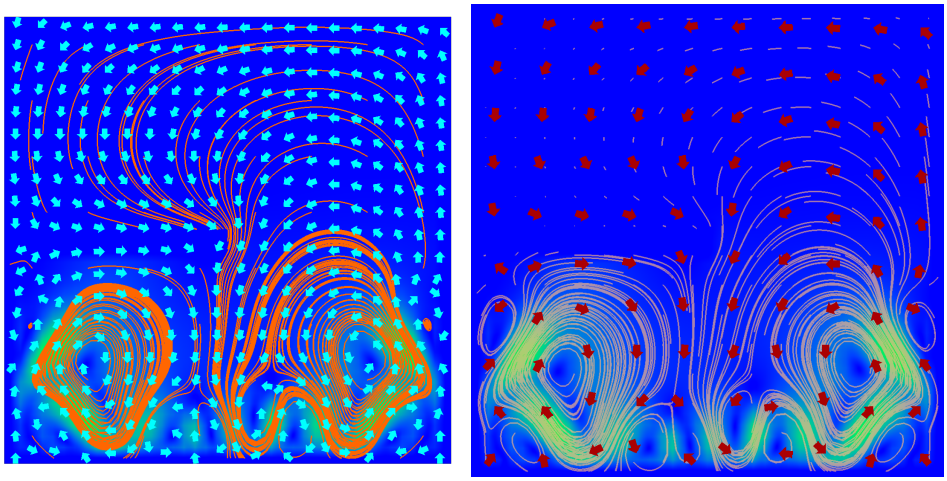
Zu dem für die vorliegende Arbeit entwickelten Programm wurde eine Expertenbasierte Evaluation durchgeführt. Die Evaluationszeit betrug pro Befragung eine Stunde. Dazu wurde das Programm mit seinen Funktionen Strömungsexperten vorgestellt, indem ihnen alle Funktionen erklärt und demonstriert wurden. Dann durften sich die Experten frei in die Funktionen des Programms einarbeiten.

Danach bekamen sie drei Aufgaben, um selbst ein, ihrer Meinung nach aussagekräftiges Bild, für drei unterschiedliche Strömungszeitpunkte zu erstellen. Nach der Bearbeitung wurde dann ein Fragebogen zu dem Programm ausgefüllt.

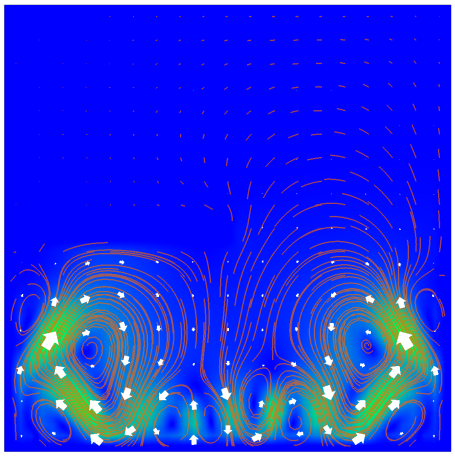
Zusätzlich zu den im Programm implementierten Strömungsvisualisierungstechniken wurden FTLE-Bilder zu den vorgegebenen Zeitpunkten zur Verfügung gestellt.

6.0.1 Aufgabe 1

Die Aufgabe war, zu einem vorgegeben Zeitpunkt die momentane Strömung so genau und übersichtlich wie möglich zu beschreiben. Die Ergebnisse sind in den folgenden Bilder zu sehen:



(a) (b)

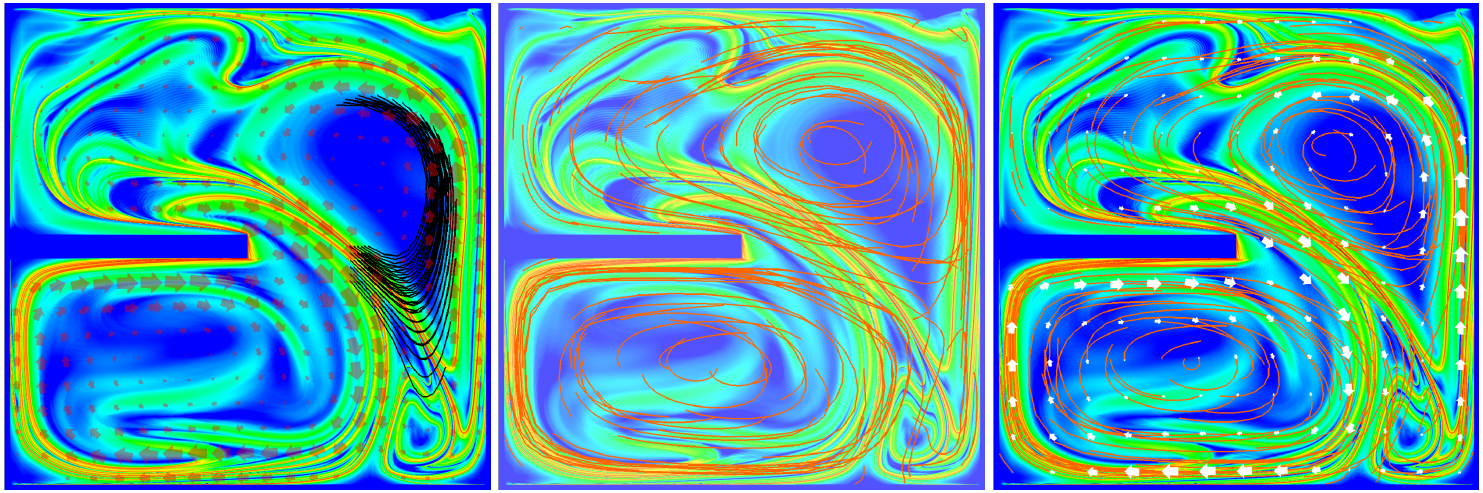


(c)

Abbildung 6.1: Bei den Bildern wurden die selben Visualisierungstechniken verwendet. Jedoch sind Unterschiede bei den gewählten Einstellungen um die Strömungsbilder zu erzeugen zu erkennen. In Abbildung (a) wurde, im Vergleich zu den beiden anderen Bildern, ein dichteres Pfeilbild gewählt, zudem überdecken die Stromlinien Teile des Farbverlaufs um die Stömungsgeschwindigkeit dazustellen. In Abbildung (b) wurden wie in Abbildung (c) die Stromlinine transparent gemacht, um den Farbverlauf nicht zu überdecken. Die gewählten Pfeile beschreiben nicht so detailliert wie in (a) in welche Richtung sich die Strömung bewegt, jedoch wird dies im unterm Bereich der Strömung durch ein dichteres Stromlininebild kompensiert. In Abbildung (c) wurden Pfeile verwendet, die zusätzlich an die Strömungsgeschwindigkeit angepasst sind. Somit wird die Strömungsrichtung im oberen Bereich der Strömung ausgeblendet.

6.0.2 Aufgabe 2

Die Probanden sollten zu einem bestimmten Zeitpunkt nach eigenem Empfinden die Strömung so genau und übersichtlich wie möglich beschreiben. Die Wahl der dabei verwendeten Techniken war dabei frei. Die Ergebnisse sind in den folgenden Bildern zu sehen:

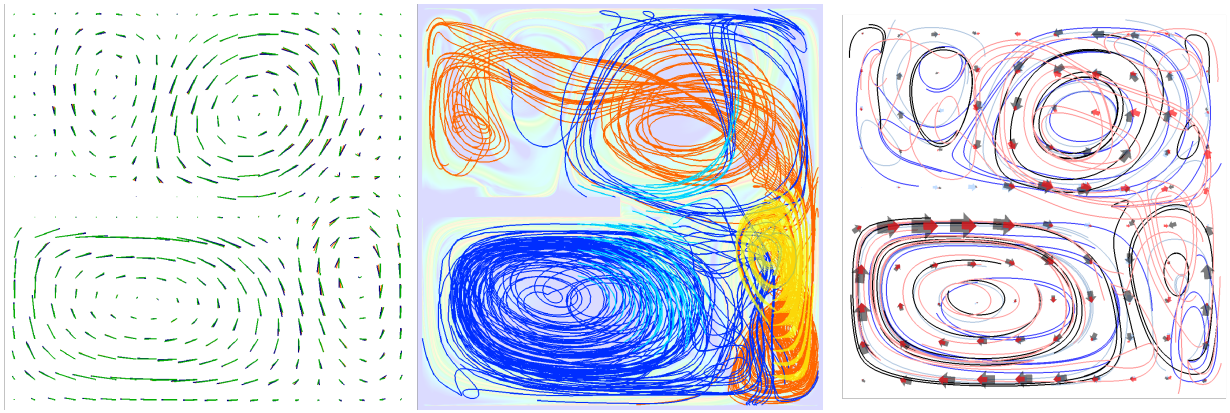


(a) (b) (c)

Abbildung 6.2: In jedem Bild wurde ein FTLE-Bild in den Hintergrund gelegt, um die Separation der Partikel dazustellen. In Abbildung (b) wurde das Bild zusätzlich Transparent gemacht, wodurch das Bild etwas matter wirkt. In Abbildung (a) wurde wie in Abbildung (c) die aktuelle Strömungsgeschwindigkeit durch Pfeile sichtbar gemacht. Im Gegensatz zu Abbildung (c) wurden die Pfeile transparent gemacht um die Informationen des FTLE-Bildes nicht zu überdecken. Dafür zeigen die Pfeile in Abbildung (c) zusätzlich die Startposition der Pfadlinien. In Abbildung (a) wurde nur ein selektierter Bereich durch die Pfadlinien beschrieben. Hingegen wurden in Abbildung (b) und (c) die Startpunkte des Pfadlinien gleichmäßig auf dem Bild verteilt.

6.0.3 Aufgabe 3

Die Probanden sollten zu einem fest vorgegeben Zeitpunkt die Strömung mit zeitabhängigen Visualisierungsmethoden so genau und übersichtlich wie möglich beschreiben.

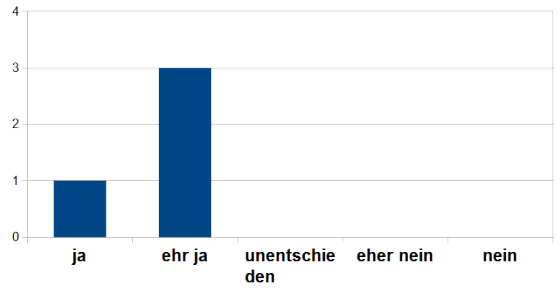


(a) (b) (c)

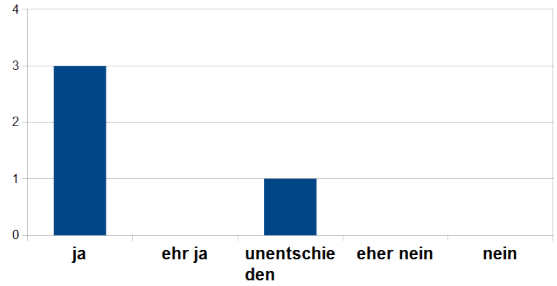
Abbildung 6.3: Bei dieser Aufgabe sind drei völlig unterschiedlich Bilder entstanden. In Abbildung (a) wurden zur Beschreibung der Strömung nur Pfadlinien verwendet. Diese Pfadlinien beschreiben jeweils den Verlauf eines masselosen Partikels für einen Zeitschritt. Die blaue Pfadlinie beschreibt den Verlauf eines Partikels vor dem Zeitpunkt, die orange zum aktuellen Zeitpunkt und die grüne zum zukünftigen Zeitpunkt. Somit kann die Änderung der Strömungsrichtung und Stärke beim Übergang und Verlasses des aktuellen Zeitpunktes mit der aktuellen Strömung verglichen werden.
In Abbildung (b) wird ein FTLE-Bild, zur Beschreibung der Separation der Partikel transparent im Hintergrund dargestellt. Zudem werden Pfadlinien in zwei unterschiedlichen Bereichen im Strömungsfeld erzeugt. Der Beginn der Pfadlinien wird durch die hellblauen und gelben Pfadlinien dargestellt.
In Abbildung (c) Wurden Pfadlinien zu regelmäßigen zeitlichen Abständen erzeugt. Zwei vor dem zu beschreibenden Zeitpunkt, zwei danach und einer direkt zum Zeitpunkt. Durch die transparenten Pfeile wird die Strömungsgeschwindigkeit und Richtung zum Erzeugungsmoment der Pfadlinien angezeigt. An den Farben ist zu erkennen, welche Pfeile zum selben Zeitpunkt erzeugt wurden wie die Pfadlinien.

6.0.4 Resultate des Fragebogens

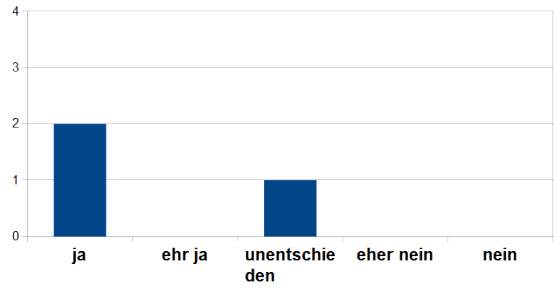
In dem von den Probanden ausgefüllten Frageboden wurden die folgenden Fragen beantwortet:



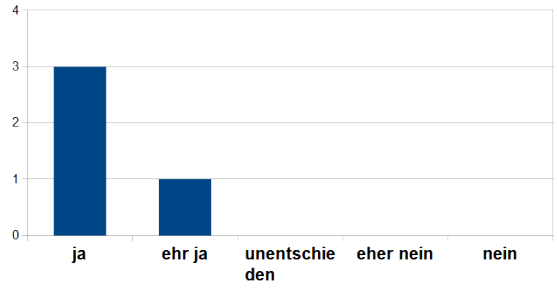
Die von der Anwendung zur Verfügung gestellten Funktionen reichen aus, um die angefallenen Aufgaben effizient zu erledigen?



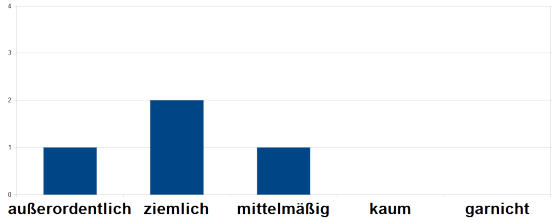
Empfanden sie die Ebenen-Hierarchie als hilfreich um die Strömungsbilder zu modifizieren?



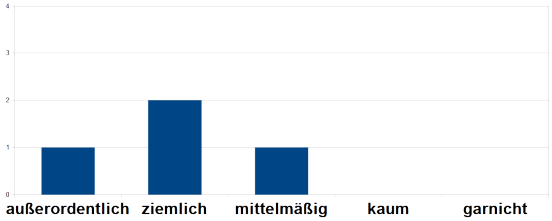
Halten Sie das Konzept Strömungsvisualisierungen mit Hilfe von z.B. Pinseln zu kombinieren für sinnvoll?



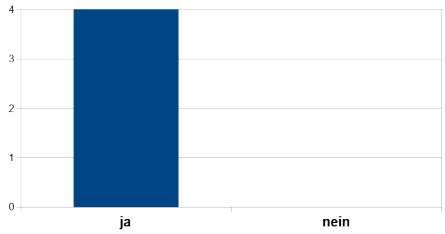
Sagt ihnen das Konzept zu, dass die erzeugten Komponenten des Gesamtbildes jederzeit nachträglich modifiziert werden können? (Wie Beispielweise das Ändern des Zeitraumes oder die Dichte des Glyphen-Feldes)



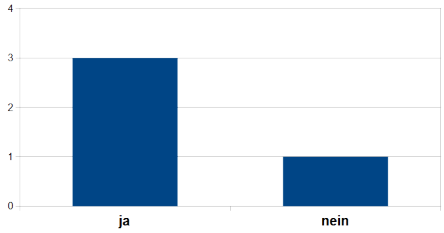
Wie stark unterstützt Sie das Programm beim erledigen ihrer Aufgaben?



Wie übersichtlich fanden sie das Programm?



Ist Ihnen die Problematik, dass die verschiedenen Strömungsvisualisierungstechniken oft nicht ausreichen um eine Strömung hinreichend zu beschrieben klar bzw. klar geworden?



Konnten sie gut mit der Ebenen-Hierarchuie umgehen?

Alle Probanden fanden es sinnvoll verschiedene Strömungsbilder zu kombinieren um Strömungseigenschaften genauer zu beschreiben da sie meinten, dass eine Technik oft nicht ausreichen würde.

Die folgenden Erweiterungsvorschläge wurden gemacht:

- Benutzer sollten die Startpunkte zur Erzeugung der Strom-, Pfadlinien nicht nur durch Setzen eines Feldes erzeugen können.
- Zusätzlich sollten Streaklines angezeigt werden.
- Masken sollten invertierbar sein
- Ein Modus um beim Zeichen, die Gesamttextur der selektierten Ebene darzustellen.

Durch die Auswertung des Fragebogens kann geschlossen werden, dass das Malprogramm alle nötigen Werkzeuge bietet, um verschiedene Strömungsmethoden sinnvoll zu kombinieren, es aber noch Möglichkeiten gibt, das Programm zu erweitern und übersichtlicher zu gestalten. Bei der Ebenen-Hierarchie könnte das verschieben der Ebenen intuitiver gestaltet werden, da die verschobene Ebene immer über die selektierte Ebene gelegt wird, und somit wenn eine Ebene auf die Position der untersten geschoben werden soll zwei Schritte nötig sind.

7 Resultate

7.1 Strömungseigenschaften bei konstanter Zeit

Im Folgenden werden Strömungsbilder erzeugt, die Informationen der Strömung zu einem festen Zeitpunkt darstellen.

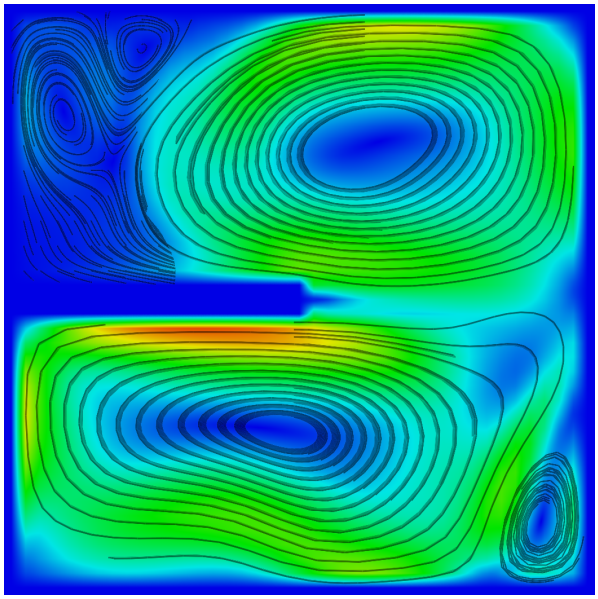


Abbildung 7.1: Mit Hilfe des Programms kann ein Bild im Gesamten transparent gemacht werden, um zusätzliche Informationen einzufügen. In diesem Bild wurden die Stromlinien transparent gemacht, damit sie den Farbverlauf nicht überdecken.

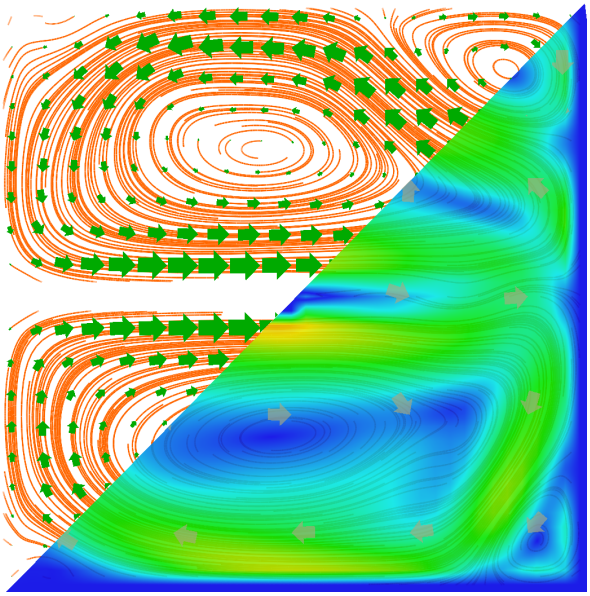


Abbildung 7.2: In diesem Bild wird das Polygon Werkzeug demonstriert. Das Resultatbild zeigt einen Vergleich Strömung auf unterschiedliche Weise darzustellen. Im linken oberen Bereich wird die Strömungsgeschwindigkeit mit Pfeilen dargestellt. Diese Technik, bietet Platz für zusätzliche Informationen. Im Gegensatz zum Farbverlauf im rechten Teil, hier kann nur durch Transparenz zusätzliche Information dargestellt werden ohne Informationen zu überdecken.

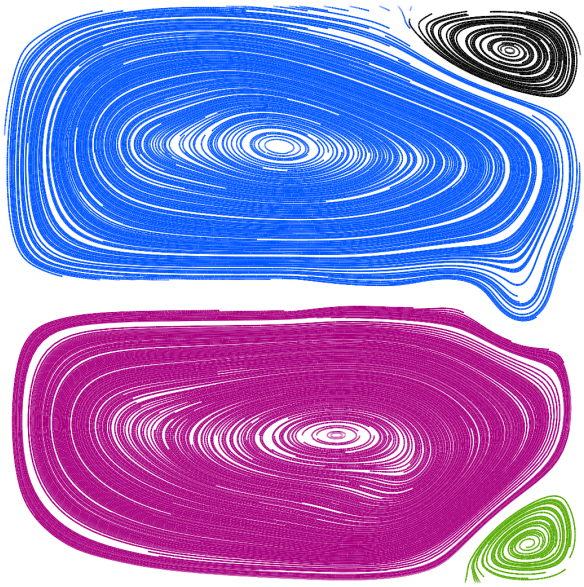


Abbildung 7.3: Die verschiedenen Wirbel der Strömung können durch das Programm unterschiedlich gefärbt werden. Somit kann der Kontrast zwischen den Wirbeln erhöht werden.

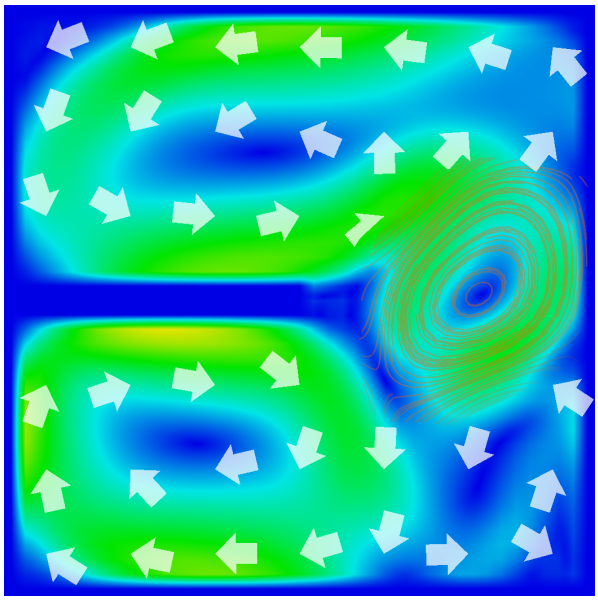


Abbildung 7.4: In diesem Bild wurde mit Hilfe des Pinsel-Werkzeuges der mittlere Wirbel hervorgehoben, um die Aufmerksamkeit des Betrachters darauf zu lenken.

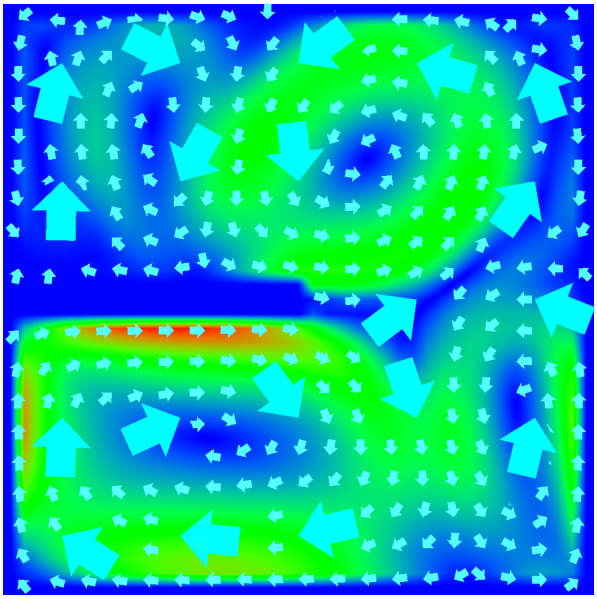
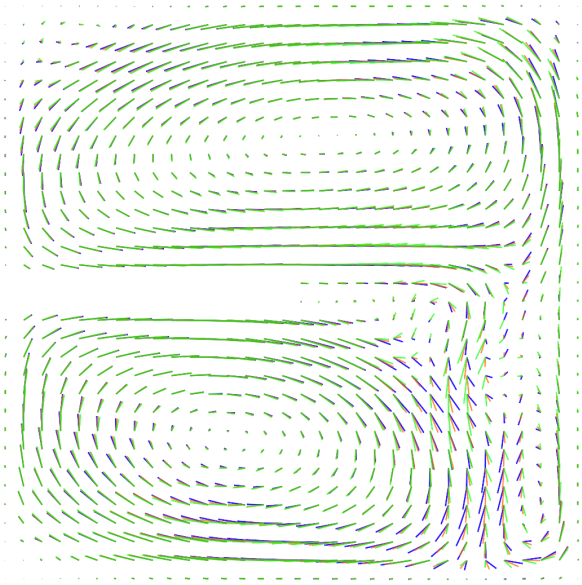


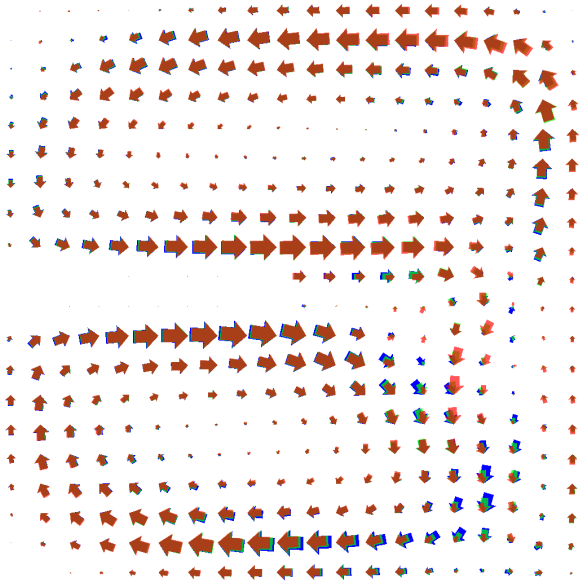
Abbildung 7.5: Mit den großen Pfeilen wird der grobe Verlauf der Strömung angedeutet, die kleinen Pfeilen beschreiben den Verlauf genauer. Somit kann der Betrachter durch den groben Verlauf der Strömung betrachten und wenn er sich für mehr Details interessiert die kleinen Pfeile betrachten.

7.2 Zeitabhängige Strömungseigenschaften

Die folgenden Resultatbilder beschreiben einen zeitlichen Verlauf der Strömung.



(a)



(b)

Abbildung 7.6: Darstellung der Strömung zu verschiedenen Zeitpunkten in (a) mit Stromlinien in (b) mit Pfeilen. Grün der vorherige Zeitpunkt, orange der aktuelle Zeitpunkt und rot der nachfolgende Zeitpunkt. Somit kann der Unterschied der Strömungsrichtung und Stärke der verschiedenen Zeitpunkte dargestellt werden.

7.3 Strömungseigenschaften allgemein

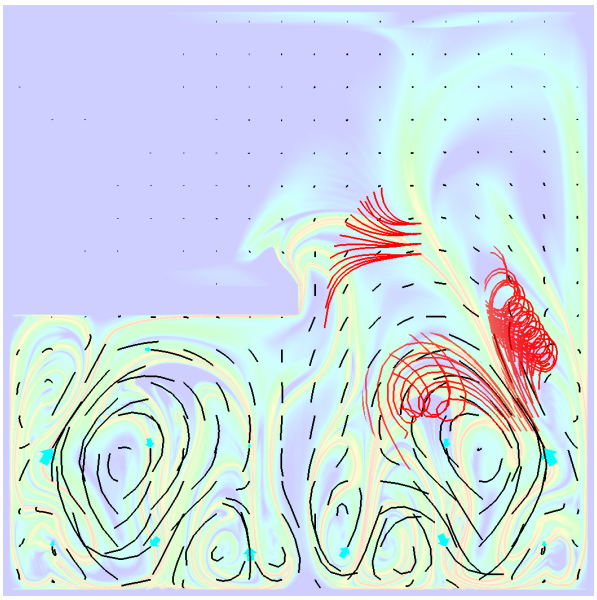


Abbildung 7.7: Darstellung der Strömung durch Verwendung eines FTLE Bildes. Schwarz einige Pfadlinien werden darstellt mit Rot wird der zeitabhängige Verlauf selektierter Bereiche hervorgehoben. Durch dieses Bild wird dargestellt, wie Strömungsbilder untersucht, beziehungsweise bestimmte Eigenschaften hervorgehoben werden können.

8 Zusammenfassung und Ausblick

Wie in der Arbeit beschrieben, ist es eine sehr komplexe Aufgabe, Informationen in einem Bild zu maximieren und dabei darauf zu achten, den Bildbetrachter nicht zu überfordern. Darum sind oft mehrere Versuche notwendig, verschiedene Bilder zu kombinieren, um dieser Aufgabe gerecht zu werden.

Aus diesem Grund, ist das in dieser Arbeit entwickelte Malprogramm für Strömungsvisualisierung entwickelt worden, um die Möglichkeit zu bieten, schnell und einfach Strömungs-Bilder zu erzeugen und zu kombinieren. Die angebotenen Kombinationswerkzeuge sind, Pinsel-, Polygon- und Lininewerkzeug. Dazu verwendet das Programm Masken um die Angewendeten Zeichenoperationen nicht direkt auf die Strömungsbilder anzuwenden. Dies hat außerdem den Vorteil, dass die angewendeten Operationen gelöscht, verschoben und kopiert werden können. Zusätzlich können die erzeugten Bilder hierarchisch dargestellt und intern verwaltet werden. Zudem ist es möglich, verwendete Bilder auszublenden um zu experimentieren, welche Bilder sich besser ergänzen.

Durch die durchgeführte Expertenbasierte Evaluation kann festgestellt werden, dass das erzeugte Programm brauchbar aber noch nicht am Ende seiner Entwicklung ist, da es noch viele Möglichkeiten bietet, erweitert zu werden.

Mögliche Erweiterungen sind beispielsweise weitere Strömungsvisualisierungstechniken oder Filter zu implementieren oder das Programm so zu erweitern, dass dreidimensionale Strömungen erzeugt und kombiniert werde können.

Literaturverzeichnis

[AN11] P. A. Nischwitz, M. Fischer, Herausgeber. *Computergraphik und Bildverarbeitung*. VIEWEG + TEUBNER, 2011. (Zitiert auf Seite 30)

[CL93] B. Cabral, L. C. Leedom. Imaging vector fields using line integral convolution. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '93, S. 263–270. ACM, New York, NY, USA, 1993. doi:10.1145/166117.166151. URL http://doi.acm.org/10.1145/166117.166151. (Zitiert auf Seite 11)

[DB07] R. M. T. I. David Borland. Rainbow Color Map (Still) Considered Harmful. *IEEE Comput. Graph. Appl.*, 27(2):14–17, 2007. doi:10.1109/MCG.2007.323435. URL http://dx.doi.org/10.1109/MCG.2007.323435. (Zitiert auf Seite 30)

[ELH85] D. A. N. Edwin L. Hutchins, James D. Hollan. Direct Manipulation Interfaces. Technischer Bericht, University of California San Diego, 1985. (Zitiert auf Seite 11)

[HG11] X. Y. Hanqi Guon, Ningyu Mao. WYSIWYG (What You See is What You Get) Volume Visualization, 2011. (Zitiert auf Seite 11)

[Hos13] Free Web Icons, 2013. URL www.fatcow.com/free-icons. (Zitiert auf Seite 13)

[HWS98] D. L. K. Han-Wei Shen. A New Line Integral Convolutation Algorithm for Visualizing Time-Varying Flow Fields. Technischer Bericht, 1998. (Zitiert auf Seite 11)

[JK09] I. H. B. R. N. H.-C. H. Jens Kasten, Christoph Petz. Localized Finite-time Lyapunov Exponent for Unsteady Flow Analysis. Technischer Bericht, Zuse Institute Berlin,Berlin Institute of Technology MB1, 2009. (Zitiert auf Seite 11)

[KB08] R. W. Kai Bürger, Jens Krüger. Direct Volume Editing, 2008. (Zitiert auf Seite 11)

[SB05] M. E. G. Stefan Bruckner. VolumeShop: An Interactive System for Direct Volume Illustration. Technischer Bericht, Institute of Computer Graphics and Algorithms Vienna University of Technology, Austria, 2005. (Zitiert auf Seite 11)

[Wij02] J. J. van Wijk. Image based flow visualization. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '02, S. 745–754. ACM, New York, NY, USA, 2002. doi:10.1145/566570.566646. URL http://doi.acm.org/10.1145/566570.566646. (Zitiert auf Seite 11)

Alle URLs wurden zuletzt am 21.05.2013 geprüft.

Erklärung

Hiermit versichere ich, diese Arbeit selbständig verfasst und nur die angegebenen Quellen benutzt zu haben.

(Sebastian Konle)