



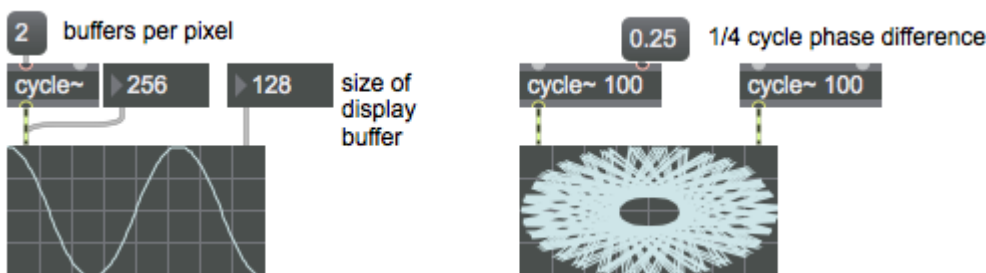
Es gibt Zeiten, in denen es aufschlussreich ist, ein Bild eines Signals zu sehen. In der realen Welt geschieht dies mit einem Oszilloskop, das eine grafische Darstellung der Amplitude über die Zeit zeigt. Max hat zwei Möglichkeiten, dies zu tun. **Die alte Schulmethode verwendet ein dediziertes Scope~ -Objekt, und die neue Schulmethode verwendet Jitter.** Das Neue hat das Alte nicht ersetzt, es gibt jeweils Anwendungen.

➔ **Beim Zeichnen eines Diagramms eines Signals in Echtzeit sind zwei Probleme zu überwinden.** Damit Ihr Auge einem zeitlich veränderlichen Signal folgen kann, muss zunächst ein Auszug des Signals erfasst und für einen bestimmten Zeitraum angezeigt werden (lang genug, damit Sie es wirklich sehen können). Daher muss das Diagramm regelmäßig angezeigt werden und bleibt immer etwas hinter dem zurück, was Sie hören. Zweitens gibt es nicht genügend Pixel auf dem Bildschirm, um eine grafische Darstellung jedes einzelnen Samples zu sehen (**zumindest nicht, ohne dass die Anzeige mit rasanter Geschwindigkeit aktualisiert wird**). Daher muss das Display ein einzelner Pixel verwenden, um viele Samples zusammenzufassen.

Alte Schule: scope~

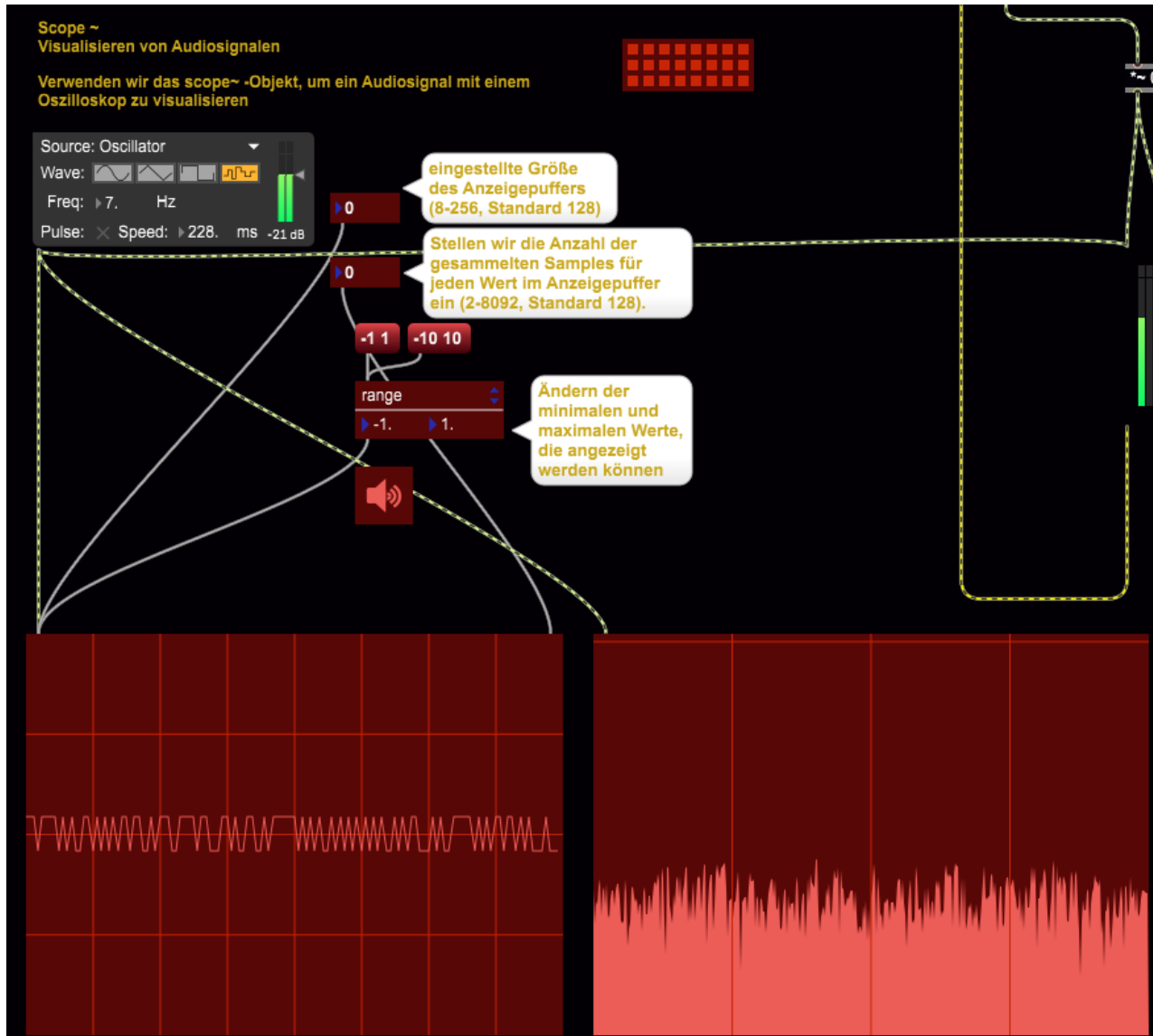
1

Das Objekt scope~ wird als rechteckige Anzeige angezeigt, die durch ein schwaches Gitter unterteilt ist. Wenn Audio eingeschaltet ist, zeigt eine horizontale Linie die Wellenform, die mit dem **linken Eingang verbunden ist**. (Diese Zeile wird als **Trace** bezeichnet.) Das Funktionsprinzip ist einfach. Eine Gruppe von Proben wird gesammelt und ein zusammengesetzter Wert wird berechnet und an einer Stelle in einem **Puffer (Zwischenspeicher)** platziert. Der Vorgang wird wiederholt, bis der Puffer voll ist. Wenn der Puffer voll ist, werden Linien von Punkt zu Punkt über die Anzeige gezogen. Die Anzahl der Stichproben pro Punkt wird durch das **Attribut calccount** festgelegt (dies kann durch ein int am linken Eingang geändert werden). Die Anzahl der Punkte wird durch das **Puffergrößenattribut** festgelegt (dies kann durch ein int im rechten Eingang geändert werden). Das Erhalten einer nützlichen Anzeige hängt von der Auswahl des **Calccounts** und der **Puffergröße** (buffer size) ab, und diese hängen davon ab, was Sie anzeigen möchten.





Tutorial Patch finden: ➔ Reference-Scope ➔ Open Help

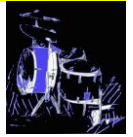


2

Anzeigen von Wellenformen

Wähle im Tutorial-Patch Sinus auf dem Umenu aus.

Wenn ein stetiges Signal an das scope~ angelegt wird, hängt das, was wir sehen, von der Beziehung zwischen Frequenz, Abtastrate, Calccount und Puffergröße ab. **Die Puffergröße sollte so hoch wie möglich sein.** Stelle daher die Breite des Objekts bis zum Maximum von 256 ein. Im Allgemeinen werden die Wellenformen umso detaillierter angezeigt, je niedriger die Anzahl ist (es wird die Wellenform effektiv heruntergesampelt) Beginne am besten bei 2. **Verwende mehr für sehr niedrige Frequenzen, aber 16 bringt uns auf 10 Hz.**



Es besteht die Möglichkeit, dass eine **stetige Wellenform nicht stetig** aussieht - sie driftet über das Display. (OK, es wird wahrscheinlich über das Display sausen!) Die Lösung für dieses Problem ist das Trigger-Attribut, das standardmäßig 0:off ist, aber **normalerweise auf 1:up gesetzt werden sollte.**

Dadurch wird die Anzeige auf einen bestimmten Wert in der Wellenform gesperrt: Triglevel. Wenn die Triggerung aktiviert ist, wartet das Objekt scope~ darauf, dass die Welle den **Triggerpegel** in der angegebenen Richtung überschreitet, bevor ein neuer Anzeigezyklus beginnt. Somit zeigt sich eine einfache Wellenform normalerweise schön und gleichmäßig. Bei einer komplexen Wellenform muss der Triglevel möglicherweise angepasst werden, um die besten Ergebnisse zu erzielen.

- Wählen Sie im Umenu ein Dreieck.
- Verwenden Sie das attrui-Objekt, um den Trigger auf up zu setzen.
- Verwenden Sie das attrui-Objekt, um den Triggerpegel anzupassen.

Beachten Sie, wie die Welle hin und her gleitet, wenn der Triggerpegel von -1 auf 1 geändert wird.

Oszillogramme anzeigen

➔ Schließen wir ein Musiksinal an das Audio-Interface an und wählen im Tutorial-Patch den Eingang auf dem Umenu.

Manchmal wollen wir das Langzeitverhalten eines Signals betrachten. Dies ist vergleichbar mit dem Verringern der Abtastrate eines Hardware-Oszilloskops. Sie **sehen eher die Hüllkurve des Signals als die einzelnen Wellen.** Die äquivalente Einstellung für scope~ ist ein hoher Wert für calccount. 128 wäre ein guter Anfang. Das erste, was Sie bemerken werden, ist, dass die Aktualisierung der Anzeige ewig dauert. Wie lange? Nun, die Anzahl der Samples, die zum Füllen des Displays benötigt werden, ist calccount * buffer size. Teilen Sie dies also durch die Sampling-Rate, um die Anzahl der Sekunden zwischen den Updates zu erhalten. **Bei einem Calccount von 128 und einer Puffergröße von 256 beträgt die Aktualisierungszeit 0,75 Sekunden.** Dieselbe Mathematik hilft bei der Schätzung des für die Anzeige einer vollständigen Wellenform geeigneten Kalibers. Sie möchten, dass die Berechnung mit der Periode der Welle übereinstimmt: eine über der Frequenz.

Was bedeuten die Gitterlinien?

Der scope~ ist durch horizontale und vertikale Linien in ein Raster unterteilt. Die horizontale Mittellinie markiert den Nullpunkt. Die anderen Horizontalen befinden sich einfach auf halber Höhe der Kanten. Die Werte oben und unten werden durch das Bereichsattribut festgelegt, das standardmäßig $\pm 1,0$ ist. Die Bedeutung der vertikalen Linien hängt von der Anzahl und der Puffergröße ab. Wie oben erwähnt, bestimmt das Produkt aus diesen geteilt durch die Abtastrate die Anzeigedauer. **Die vertikalen Linien teilen dies durch 8. Wenn also calccount 2 und die Puffergröße 256 beträgt, beträgt die Anzeigeperiode 0,0116 bei 44,1 kHz.** Die Umkehrung davon gibt die Frequenz einer Wellenform an, die das Fenster genau ausfüllen würde, sodass 86,1 Hz gerade passen sollten. Das durch die vertikalen Linien dargestellte Zeitintervall beträgt ca. 1,5 ms.

X-Y-Anzeige

F:\nur_MSP\Analyse_MSP\Oscilloscopes & Spectroscopes\Oscilloscope and Spectroscope_UM.doc



➔ Schließen wir ein Stereomusiksignal an Ihr Audio-Interface an und öffnen Sie das X-Y-Subpatch.

Wenn Sie ein Audiosignal an den rechten Eingang des Oszilloskops anschließen, **wird dieses Signal verwendet, um die X-Position (von links nach rechts) jedes Punkts zu bestimmen.** Wenn das gleiche Signal mit links und rechts verbunden ist, erhalten Sie eine diagonale Linie von links unten nach rechts oben. Wenn das Signal rechts eine phasenverschobene Version des linken ist, ist die Linie in die andere Richtung geneigt. **Wenn ein Stereosignal angeschlossen ist, ist das Ergebnis eine Art Scribble, und Sie können die Phasendifferenz von links und rechts anhand der Breite und Neigung der Form abschätzen.**

Wenn Sie geometrische Wellenformen mit beiden Einlässen von scope ~ verbinden, können Sie einige interessante, sich entwickelnde Formen erzeugen. Diese Formen werden ➔ **Lissajous-Figuren** genannt und sind an sich schon eine Erkundung wert. Die scope~ help-Datei enthält einige Beispiele.

Neue Schule: jit.catch ~ und jit.graph

Wir können Wellenformen auch in Jitter mit den Objekten jit.catch~ und jit.plot anzeigen. Jit.catch~ nimmt Samples aus einem Audiosignal und packt sie in eine eindimensionale Matrix. Die Anzahl der Abtastwerte pro Matrix wird durch das Framesize-Attribut festgelegt und hat genau die gleiche Bedeutung wie die Berechnung der Puffergröße von calccount * in scope~. Jit.catch~ hat mehrere Betriebsarten, aber die, die uns interessiert, ist Modus 3, der oszilloskopartig ausgelöst wird. Natürlich erfordert jit.catch ~ eine Reihe von Pony, um die Ausgabe von Matrizen zu veranlassen.

4

Jit.graph zeigt Daten in eindimensionalen Matrizen in einem zweidimensionalen Array von Zeichen an. Verbinden Sie das zwischen jit.catch~ und einem jit.pwindow, und Sie haben eine Wellenformanzeige. Jit.graph verfügt auch über Modi, die den Anzeigestil bestimmen:

- 0: Punkte
- 1: Linien
- 2: Fläche
- 3: Bipolarer Bereich
- 4: Balken

Linien sind am umfangreichsten.

Das Schwellenwertmerkmal in jit.catch~ ist insofern etwas seltsam, als der Schwellenwert immer in der Mitte der Matrix platziert wird. Wenn Sie also die Bildgröße anpassen, bleibt die Wellenform in der Anzeige zentriert. Das bedeutet auch, dass die Triggerrichtung (Trigdir) im Down (1) -Modus eingestellt werden muss, um dem Umfang zu entsprechen, den ~ mit derselben Welle macht. Die Anzeige kann mit den Attributen rangelo und rangehi weiter geändert werden, mit denen die Werte festgelegt werden, die am unteren oder oberen Rand der Anzeige angezeigt werden. Es gibt auch ein Höhenattribut, das die vertikale Abmessung der Matrix festlegt. Dies sollte so eingestellt werden, dass es der Höhe des jit.p-Fensters entspricht, um Größenkonvertierungsartefakte zu vermeiden.



MSP Analysis Oscilloscope and Spectroscope

Die Wahl zwischen Scope ~ oder Jitter zum Anzeigen von Audio hängt davon ab, was Sie tun möchten. Wenn im Patch eine Audioanzeige benötigt wird, ist der Bereich ~ normalerweise die beste Wahl.

Wenn Sie das Bild à la iTunes weiterverarbeiten möchten, ist Jitter der richtige Weg.

Ein Spektrum zeichnen

Zwei beliebte Schlagworte in der DSP-Community sind Zeitbereich und Frequenzbereich. Zeitbereich bedeutet einfach, dass wir grafisch darstellen, wie sich Signalwerte im Laufe der Zeit ändern, wie wir oben gezeigt haben. Der Frequenzbereich zeigt Audio zu einem bestimmten Zeitpunkt an, zeigt jedoch den Fourier-Frequenzinhalt. Somit hat der Graph (Spektrogramm genannt) eine Frequenz über den Boden. Sie werden sich an das Fourier-Theorem erinnern, das besagt, dass jede Wellenform in eine Summe von Sinuswellen dekonstruiert werden kann. Ein Spektrogramm zeigt die Amplitude dieser Sinuswellen. Wir werden in Analyse Kapitel 4: Einfache Fourier-Analyse detailliert darauf eingehen. Hier werden wir nur eine einfache Möglichkeit untersuchen, ein Spektrogramm anzuzeigen.

5

Spektrogramm

→ Wählen Sie im Umenu ein Dreieck und stellen Sie die Frequenz auf 440 Hz ein

Das Spektroskop~ -Objekt zeigt sofort ein schönes Spektrogramm an. Die Peaks in der Kurve zeigen die Frequenz und Amplitude der Teiltöne in der Dreieckswellenform. In einem idealen System wären diese Peaks dünne Linien, aber der Algorithmus, der erforderlich ist, um ein Spektrum in Echtzeit anzuzeigen, erzeugt 'Röcke' am unteren Rand der Peaks. Das wichtigste zu beachtende Attribut ist die Frequenzachsenskalierung, die vom logfreq-Attribut ausgewählt und im Attrui als Anzeige der Frequenzachsenskala angezeigt wird. Im linearen Modus ist der gesamte hörbare Bereich von 20 bis 22 kHz von links nach rechts gleichmäßig verteilt. Das Spektroskop~ -Display sollte einen schönen Satz gleichmäßig verteilter Harmonischer (Obertöne) zeigen. Es ist eigentlich jede andere Harmonische, wie Sie durch Auswahl eines Rechtecks sehen können. Wenn Sie die Eingabe auswählen, wird der größte Teil der Aktion auf der linken Seite angezeigt. Das liegt daran, dass Musik meist auf einen relativ niedrigen Frequenzbereich beschränkt ist.

→ Wählen wir in dem am Spektroskop~ -Objekt angehängten Attribut die Option Frequenzachsenskala anzeigen und dann Logarithmische Skala.

→ Wählen Sie im Umenu den Sinus und stellen Sie die Frequenz auf 43 Hz ein.

Im logarithmischen Modus sind die Oktaven gleichmäßig über die Anzeige verteilt. Das Display sollte etwa ein Drittel des Weges einen breiten Peak anzeigen. (Ignorieren Sie, was sich links vom Peak befindet. Die Bildschirmauflösung beträgt nur 43 Hz, also alles, was darunter liegt, ist Müll.) Ändern Sie nun die Frequenz auf 86 Hz und Beachten Sie, wie sich der Peak bewegt. Wenn Sie die Oktaven erhöhen: 129, 172, 215 usw., bewegt sich der Peak um den gleichen Betrag. Der Gipfel wird



MSP Analysis Oscilloscope and Spectroscope

auch schmaler. Das 43-Hz-Auflösungsband ist weniger, wenn wir uns nach oben bewegen.

→ Wählen wir nun die Eingabe.

Die Musik füllt das Display gut aus.

Beide Anzeigemodi sind sehr nützlich: Der lineare Modus zeigt die harmonische Struktur von stetigen Wellenformen und der logarithmische Modus zeigt die Klangbalance von Musik auf intuitive Weise.

Sonogramm

Durch Auswahl des Sonogrammmodus können wir sowohl Zeitbereichs- als auch Frequenzbereichsinformationen in einer Anzeige abrufen.

→ Wählen Sie im Attrui den Anzeigemodus und dann Sonogramm.

→ Wählen Sie im Umenu den Eingang.

Sie sollten jetzt ein Bild sehen, das mit schwarzen, grauen und weißen Streifen gezeichnet ist. Ein Sonogramm ist ein zweidimensionales Diagramm, das die Zeit unten und die Frequenz von unten nach oben zeigt. Die Amplitude an jedem Punkt wird durch die Dunkelheit der Farbe dargestellt. Während Sie die Musik hören, werden Sie sehen, wie die tiefen und hohen Noten angezeigt werden, und sogar die harmonische Struktur des Klangs sehen. (Der Logfreq-Modus wirkt sich auch auf Sonogramme aus.) Wählen Sie für eine besondere Behandlung die Sonogrammfarbe im Attrui aus und wählen Sie die Farbe. Jetzt zeigt sich die Intensität der Frequenzkomponenten als Rot, das durch Gelb und Grün nach Purpur verblasst.

6

Das Spektroskop~ -Objekt kann auf verschiedene Arten angepasst werden, einschließlich vertikaler Ausrichtung, Zeichen- oder Bildlaufmodus, umgekehrter Richtung und Farben. Sie können auch den anzuzeigenden Bereich einschränken, um in einem kleinen Zeitraum mehr Details zu erhalten.

Zusammenfassung

Das Objekt scope~ gibt eine Oszilloskopansicht eines Signals an und zeichnet die Amplitude über die Zeit grafisch auf. Wellenformen können auch im Jitter-Matrix-Format mit jit.catch~ und jit.graph angezeigt werden. Das Spektroskop-Objekt kann den Frequenzinhalt entweder als dynamisches Spektrogramm oder als bewegliches Sonogramm anzeigen.

Siehe auch

Name	Description
scope~	Visualize an audio signal
jit.catch~	Transform signal data into matrices
jit.graph	Perform floating-point data visualization
spectroscope~	Signal spectrogram or sonogram
MSP Analysis Tutorial 1: Signals and Meters	
MSP Tutorials: Table of Contents	
MSP Analysis Tutorial 3: Using the FFT	

Check out projects made by other Max users

F:\nur_MSP\Analyse_MSP\Oscilloscopes & Spectroscopes\Oscilloscope and Spectroscope_UM.doc