

アイデア概要

タイトル：「DemocraSea ～魚論調査～」

チーム名：kbylab2017

エントリー番号：6

要旨

多くの人が飼うペットの中に魚がいる。魚はデリケートな生き物であり、水槽内の環境には気を使わなければならない。しかし、犬や猫などとは異なり意見を伝えることはないため、我々が彼らの意見を察することは困難である。我々が提案する「DemocraSea ～魚論調査～」は、動物の本能的な「群知能」「集合知」という性質を利用して、飼い主を超えた魚のコミュニティを構築し魚の世論（魚論）を投票により調査し、水流や餌、水槽の背景の意見を水槽内に反映させる。これにより、魚が住み心地の良い環境を構築する。また、飼い主は己が想定していなかった魚の群の意見が水槽内に反映されることで新しい飼育の方法を見つけることができる。

問題提起

魚には飼い主に意見を伝える手段がほぼなく、飼い主も魚の意見を理解できない。そのため、飼い主の独断で餌や水槽内のオブジェクト、水槽の壁など背景画像を変更しているだろう。しかし、それは魚にとって望んだ環境だろうか。飼い主には、魚が快適に生活できる環境を整えることが求められているが、自身だけで得られる情報には限りがある。

解決策

この問題を解決するために、我々は魚の投票により水槽内の環境を決定するシステムとネットワークを提案する。水槽の環境（背景など）について飼い主が提案し、魚はその提案について投票をする。投票された結果に応じて水槽内の環境を変更する。提案と投票は一定時間ごとに行われ、次の投票が始まるまで、魚が選択した環境に水槽が変化する。また、このシステムはネットワークにより接続されている（図1）。各飼い主が水槽の環境を提案し、魚の投票で選択する。このシステムにより飼い主は魚にとっての快適な環境を模索することが可能になる。また、ネットワークされた他の水槽を見ることも可能である。

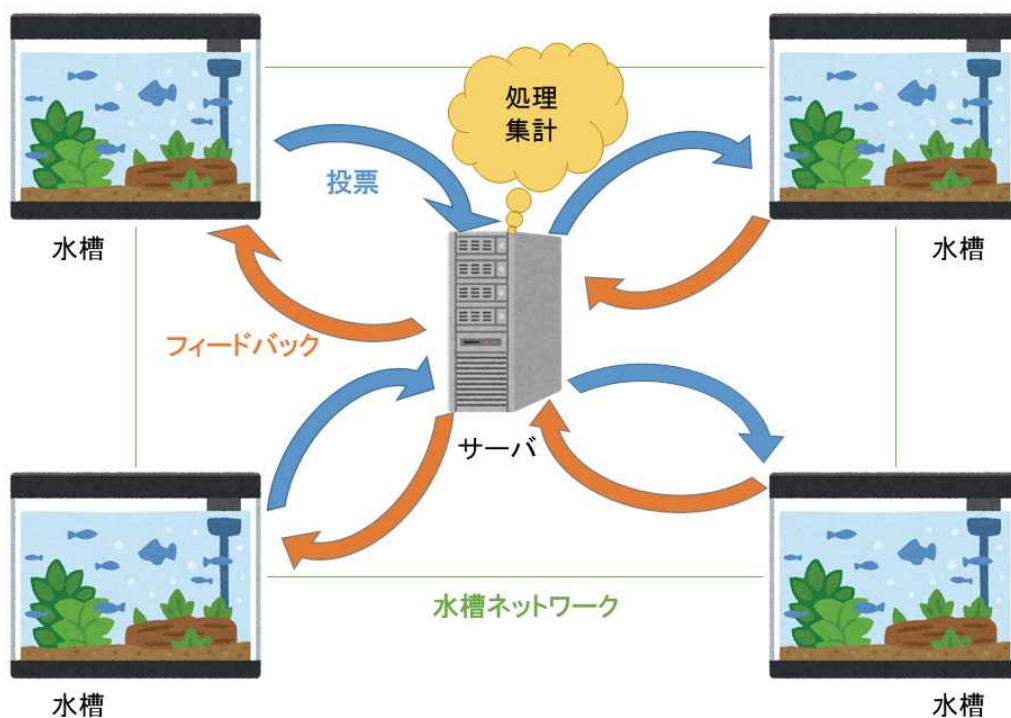


図1. 水槽ネットワークのイメージ

解決策の根拠

魚が投票を行うことで水槽内の環境を変えるこのシステムを用いることで、本当に魚の意見が反映されているのかということについて、魚は己の環境を良くしようと思えるほど賢くないのではないかと疑問を持つ方も多いと思われる。しかし、「DemocraSea ～魚論調査～」は動物の本能的な性質「群知能（一つ一つの個体が貧弱であっても群を作り協力することで仕事を成す）」と「集合知（群の多数意見は結果的に良い結果になる）」という性質を利用して、本能的に「動物が使える」を意識した設計となっている。今回、このシステムではネットワークで繋がった複数の水槽を仮想的な群として魚論調査を行う、例えば水槽の背景を選ぶとき、1体の知能レベルは低いですが群が集まる背景のところは群全体の総意から居心地が良いという判断をしたということになる。このように群による集合知により全体が良いとする環境へ変化させることができる。

参考： 今福道夫(1992) 生物集団における群知能-動物集団の知能行動

James Surowiecki(2006) 「みんなの意見」は案外正しい（角川文庫）

問題解決をすることで得られる恩恵

1. 魚のメリット

- 群の集合知を用いた魚論調査によって魚にとって住み良い水槽環境の実現ができる。
- 水槽内の群だけに作用するだけでなくネットワークに繋がった他の家の水槽で飼われている群と共同で魚論調査を行うことにより、大きなコミュニティとして作用する。

2. 飼い主のメリット

- 飼い主が想定していなかった魚の意見を水槽内に反映させることで、新しい魚の飼い方を実現する。
- 自分の考えた水槽の環境が認められることで承認欲求を満たすことができる。

解決策の実現方法

以下にソフトウェア・ハードウェアの視点からの実現方法を示す。

ソフトウェアの仕様

・システムの流れ

システムの流れを図2に示す。

1. 待機

水槽内の環境は初期の環境、または前回の投票によって選ばれた環境で待機状態とする。

2. 設定

ユーザが水槽内の環境の設定をWebアプリケーション上で行う、環境とはLEDのパターン、背景画像、水流のパターン、えさの種類を合わせたものとする。

3. 投票

待機状態からある一定の時刻が経過すると、水槽内を仮想的に左右に二分割し、2つの環境を作り魚論調査を行うための投票フェーズへ移行する。環境は飼い主から提案されたものをトーナメント形式で1対1で投票を繰り返す。トーナメントの初回はサーバ上に蓄積された各ユーザの提案した環境がランダムで2つ選ばれ投票が行われる。2回目以降、前回の投票で選ばれた環境とサーバ上に蓄積された環境がランダムで1つ選ばれ投票が行われる。投票を繰り返し、最後に残った1つの環境を全体に反映させる。

4. 反映/通知

投票が終了したのち、投票によって選ばれた環境を各水槽に反映する。投票結果は飼い主の持つアプリケーションに通知される。

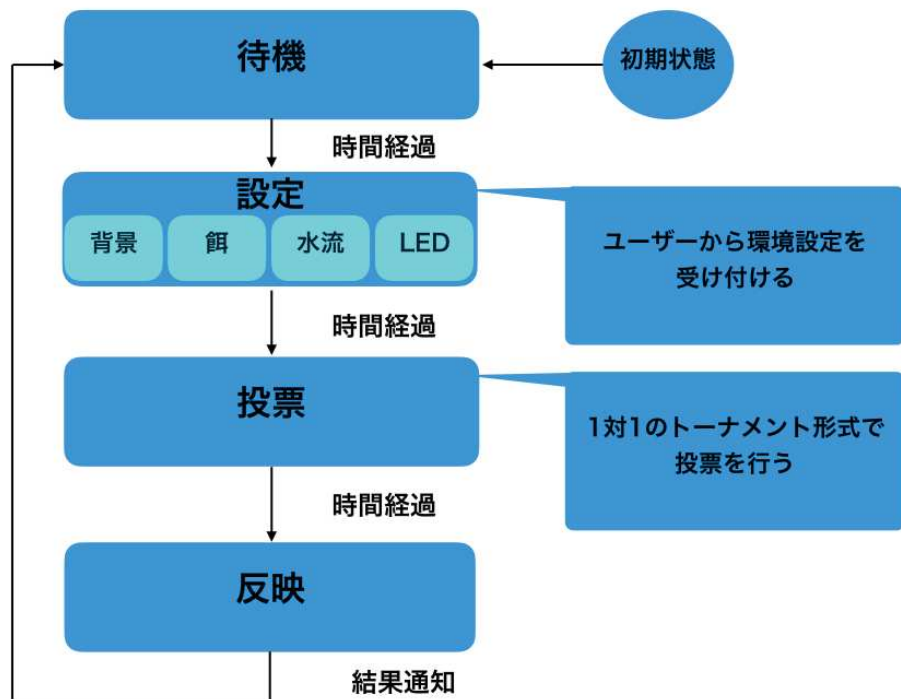


図2. システムの流れ

・実装予定の機能

1. 設定

ウェブブラウザ上のアプリケーションで水槽の環境に関する以下の設定を行う（図4a, b）。

- LEDの色や光り方のパターン
- ディスプレイに表示する水槽の背景
- 水流の強さの時間間隔のパターン
- えさの種類

2. 投票

水槽の左右には環境構築を行うための、餌やり機、LEDライト、水流モータがなどの全く同じ装置が独立に配置されており、提案された設定を右と左で同時に再現することで投票領域として機能させる。

魚が2つの環境のうちどちらを選んだか、画像差分を利用して判定する。投票のイメージを図3に示す。

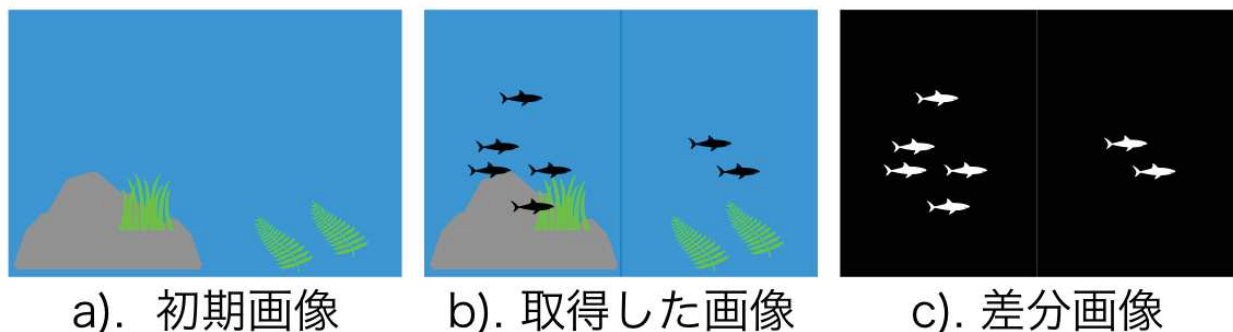


図3. 投票のイメージ

初期値として図3aに示すような魚のいない水槽の画像を作り、取得した投票中の画像から差分を取得し差分のピクセル数を数えてどちらがより多く魚がいるかを判定する。

3. 通知

ユーザの持つアプリケーションに投票結果を通知する。アプリケーションのUIイメージを図4に示す。

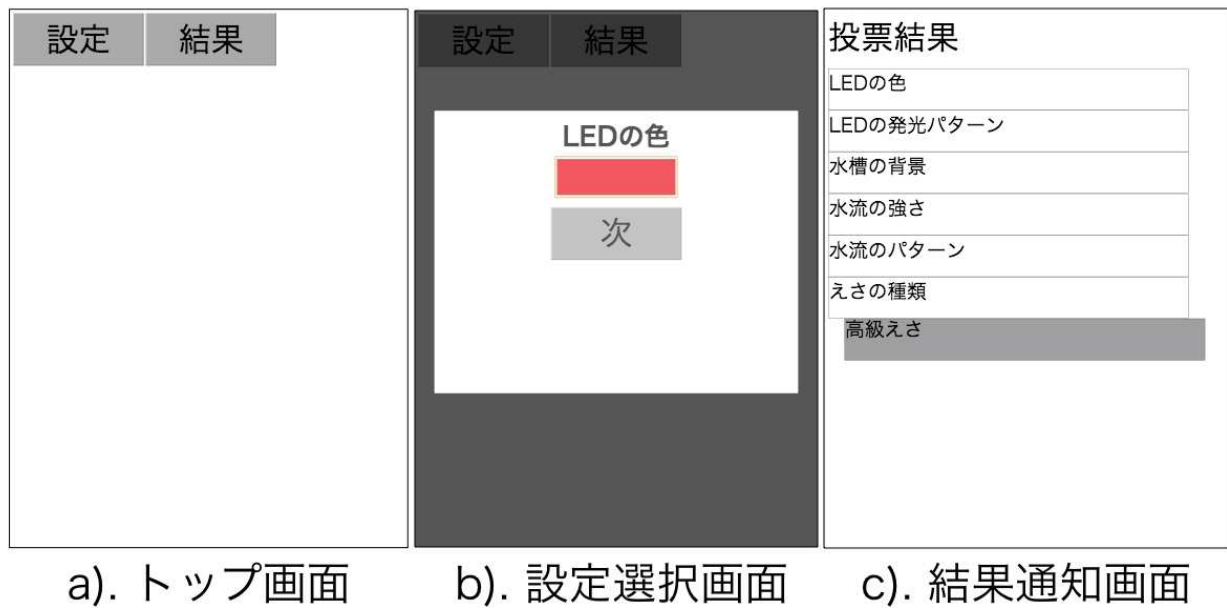


図4. アプリケーションのUIイメージ

4. ネットワーク

クライアント（各水槽を制御しているRaspberry Pi）はサーバとSocket通信とWebSocket通信を行う。また、サーバと通知アプリケーションはWebSocket通信を行う。ネットワークのイメージを図5に示す。

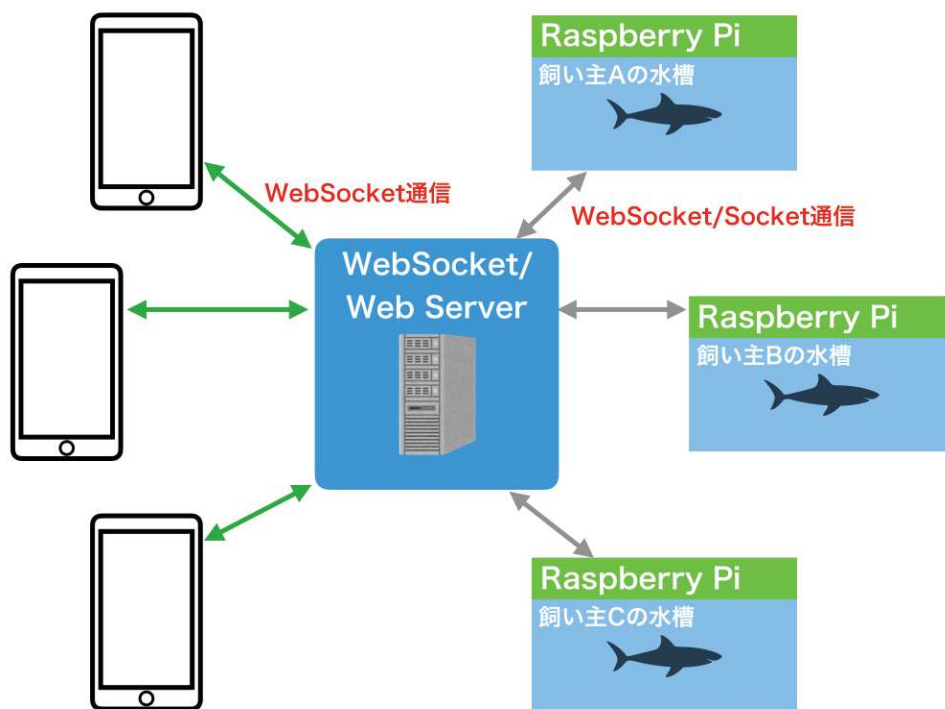


図5. ネットワークのイメージ

・使用技術

1. JavaScript

- ・ Webアプリケーション実装言語

2. Python

- ・ ディスプレイやモータ、カメラの制御を行うRaspberry Piの実装言語
- ・ 魚の多数決を判定する画像差分をOpen CVを利用する

ハードウェアの仕様

ハードウェアの構成要素を図7に示す。

1. Raspberry Pi

Raspberry Pi3を使用する。Debian系のLinuxOSであるRaspbianを動作させる。

各水槽においてハードウェアの制御、サーバとの通信を行う。

2. モニタ

背景画像の出力に使用する。Raspberry PiのHDMI端子に接続する。

3. 全天球カメラ

株式会社リコーの全天球カメラThetaを使用する。防水ケースを使用して水槽内の動画撮影を行う。

通常のカメラを用いた場合、水槽の全景を撮影するためには水槽から離れた位置にカメラを設置する必要がある。設置場所に関する制約が大きくなる。今回は全天球カメラを水槽に沈めることで、通常のカメラを用いたものよりコンパクトなシステムを目指す。

Thetaはライブストリーミングモードを用いることで一般的なUSBWebカメラとしてRaspberry Piに認識され、動画を取得することができる。ffmpegを用いて動画の録画、投票判定用画像の動画からの切り出しを行う。

4. 餌やり機

図6に餌やり機のイメージ画像を示す。サーボモータ2の軸を回転させ、えさを入れる箱を持ち上げ、えさを投入する。えさは、3種類から選べるよう、3つの箱を用意している。持ち上げる箱を変えたい場合は、サーボモータ1の軸を、左右に回転させる。それにより、持ち上げる箱を変える。サーボモータの角度制御にはRaspberry PiのGPIO端子を用いたPWM制御を使用する。モータの駆動電源には乾電池を使用する。

5. 水流ポンプ

水槽内の水流の発生に使用する。市販の模型用防水モータに改造を加えたものを用いる。

モータの回転速度制御にはRaspberry PiのGPIO端子を用いたPWM制御を使用する。防水モータは餌やり機のサーボモータと異なり内部に制御回路を持たないため、別途DCモータドライバを用意して、制御を行う。モータの駆動電源には乾電池を使用する。魚と回転部分の接触を避けるため、モータ周辺に魚よけを取り付ける。

6. 照明

フルカラーLEDモジュールを使用する。Raspberry PiのGPIO端子を用いた制御を行う。

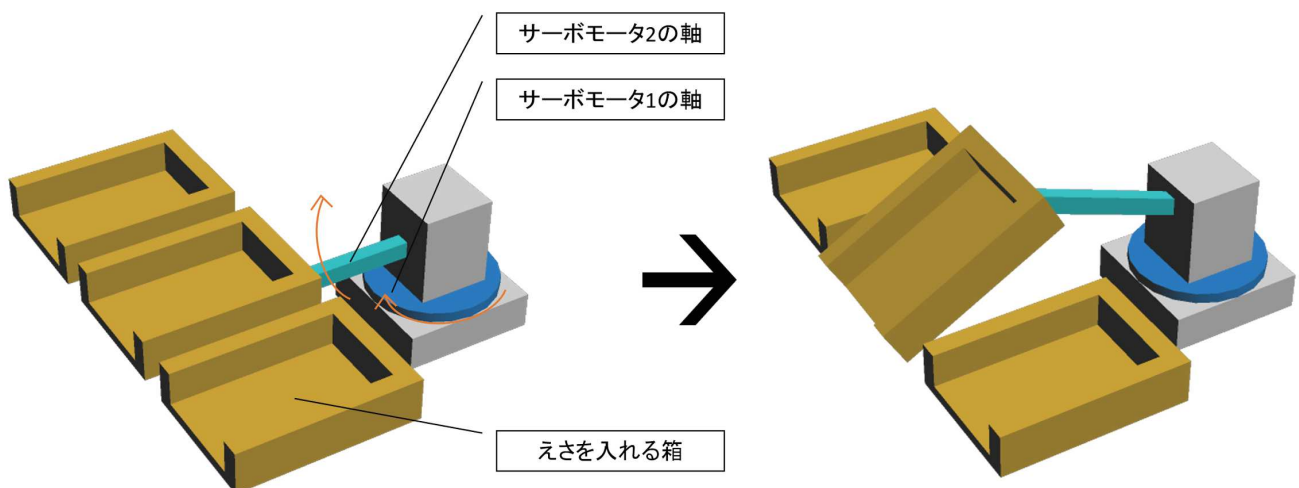


図6. エサやり機イメージ

ハードウェアの構成について図7に示す。

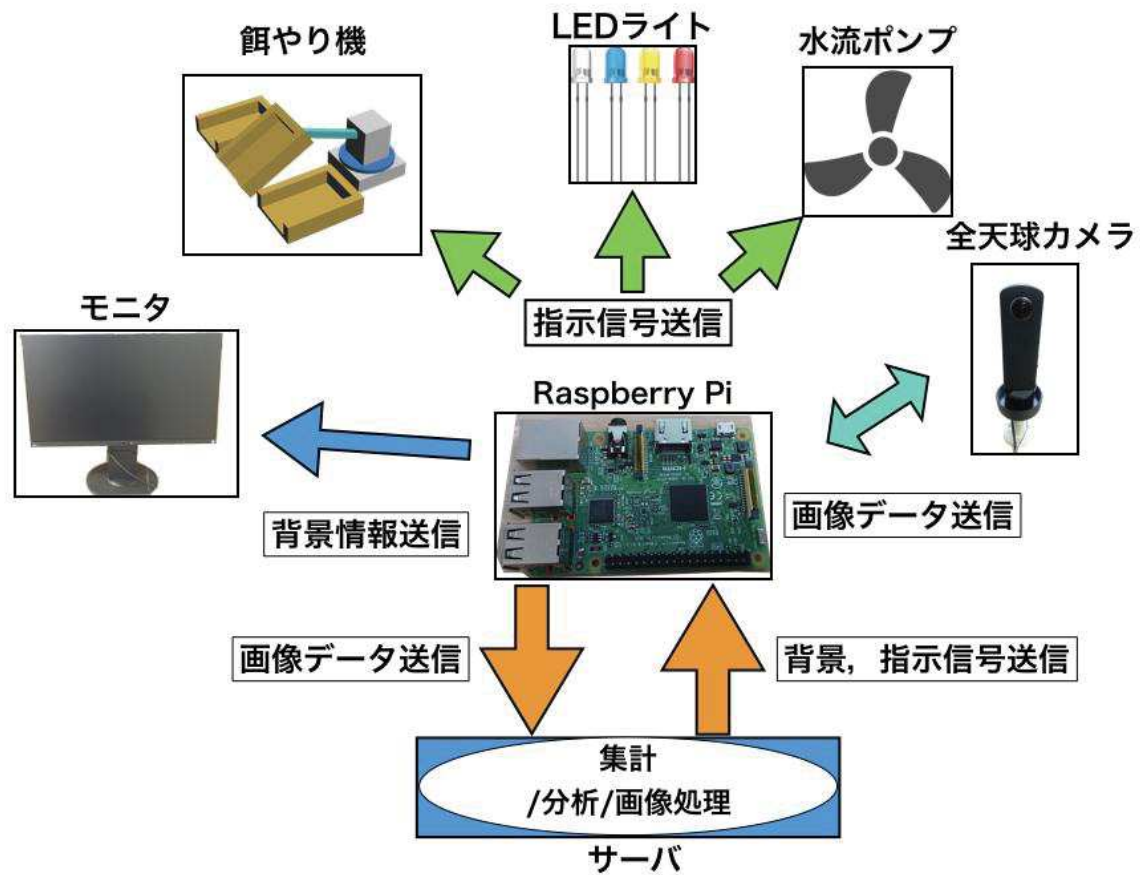


図7. ハードウェアの構成

水槽の構成図を図8に示す。

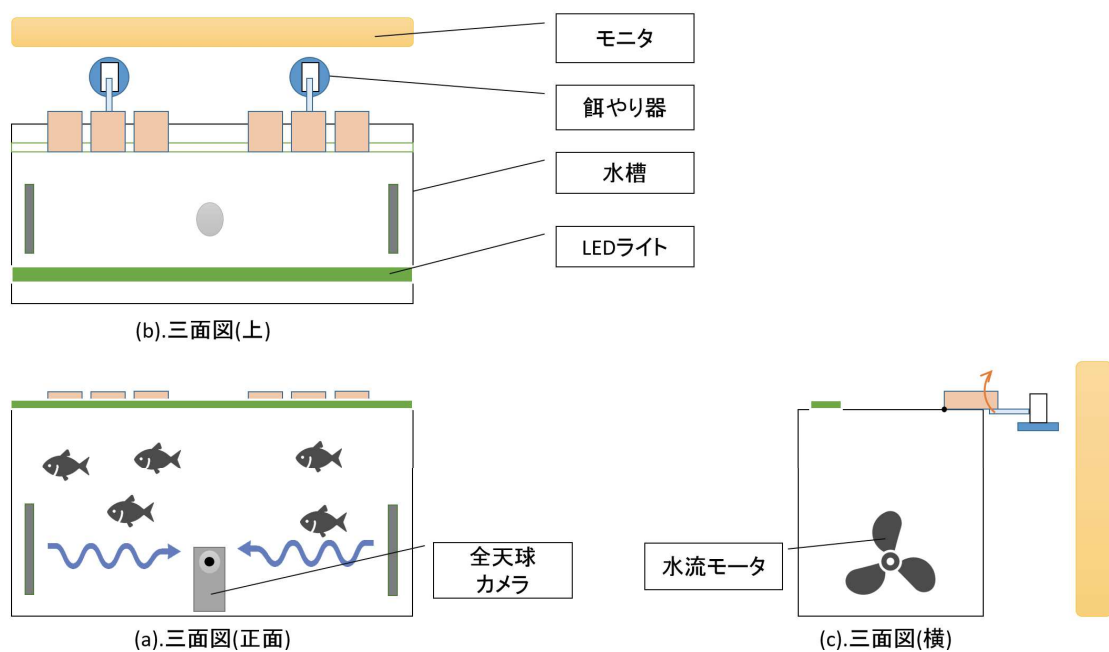


図8. 水槽の構成