

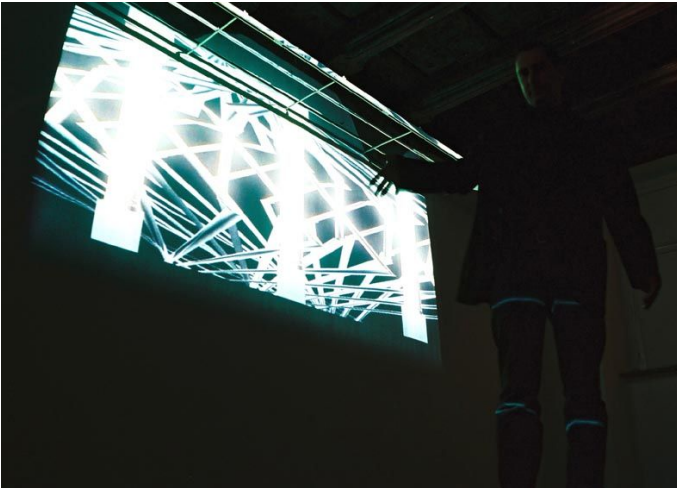
# SKANER

instalacja interaktywna



**Bartosz “Kostia” Jakubicki** – idea, animacja 3D, aranżacja przestrzeni  
**Paweł Janicki** – programowanie, muzyka

3	WPROWADZENIE (english version available)
4	IDEA
6	INTERAKCJA
9	WIZUALIZACJE
11	DŹWIĘK
12	MEDIA MODULARNE: PRZEGLĄD OPROGRAMOWANIA
15	SPRZĘT I OPROGRAMOWANIE
16	AUTORZY (english version available), PODZIĘKOWANIA

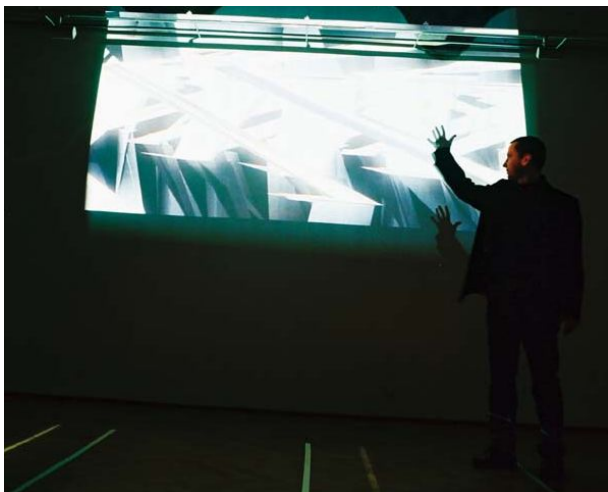


*SKANER: połączenie motion tracking, interaktywnych projekcji i dźwięku.*

SKANER jest instalacją interaktywną wystawioną w dniach 06-08.11.2003 we wrocławskiej Galerii Entropia (<http://www.entropia.art.pl>) w ramach drugiej edycji Wrocław Industrial Festival (<http://jobkarma.terra.pl/festival>). W niniejszym artykule zamierzam omówić ideę instalacji oraz technologie i metody jej realizacji. Mam nadzieję, że tekst ten przybliży wszystkim zainteresowanym praktyczną stronę pracy nad tego rodzaju przedsięwzięciami oraz pomoże artystom i producentom, zwłaszcza początkującym, w realizacji ich własnych zamierzeń. Tekst nie jest

natomiast krytyczną analizą instalacji.

*This article is a description of the SKANER project: interactive, place specific, multimedia installation designed as a part of the Wrocław Industrial Festival (<http://jobkarma.terra.pl/festival>), exhibited 06-08.11.2003 in Entropia Gallery (<http://www.entropia.art.pl>). Article is not a kind of critical essay, but is devoted technical and logistic aspects of creating interactive installations and environments, especially building software, real-time audio and video processing, motion tracking systems.*

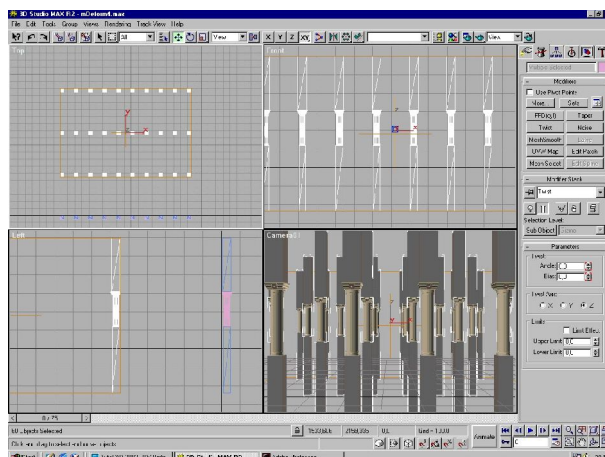


SKANER we wnętrzach Entropii.

Jako, że Kostia zajmuje się eksplorowaniem tematyki związanej z rzeczywistością mieszaną, szczególnie w kontekście architektury, koncepcja instalacji bazowała na wykorzystaniu istniejących fizycznie elementów architektonicznych: stworzeniu ich cyfrowych reprezentacji a następnie przywróceniu światu fizycznemu, przede wszystkim w formie projekcji wideo. Zamiarem Kostii było uzupełnienie przestrzeni galeryjnej o elementy generowane komputerowo, których forma uzależniona byłaby od aktywności publiczności znajdującej się w pomieszczeniu, jednocześnie zaś obiekty te miały być zestawione z ich fizycznym

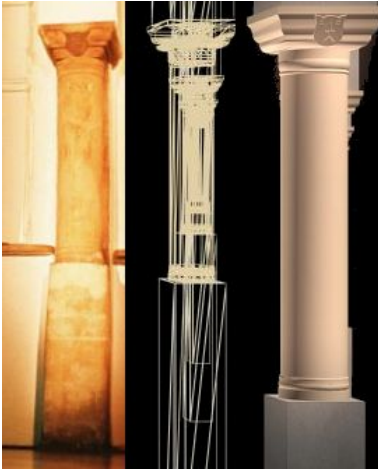
pierwowzorem.

Taka koncepcja wymagała znalezienia fizycznie istniejącego, nadającego się do digitalizacji obiektu, najlepiej istniejącego już w przestrzeni udostępnionej przez Entropię. Kostię zainspirowała niewielka kolumna stojąca w galerii; tworzył jej model w programie 3D Studio Max. Używając modelu wygenerował następnie serię krótkich, trwających 3 sekundy każda, animacji, przedstawiających rozmaite przekształcenia i deformacje. Każda z animacji mogła być odtwarzana jedno- lub wielokrotnie (w pętli) oraz łączyć się płynnie z dowolną inną (innymi słowami: wszystkie animacje zaczynały się i kończyły identyczną klatką obrazu).



3D Studio Max: narzędzie, którym posłużył się Kostia do budowy modeli i animacji 3D.

Kostię interesowała możliwość interaktywnego sterowania projekcją. Swoje animacje podzielił na trzy grupy dzieląc je pod względem charakteru i intensywności przekształceń, chciał by odtwarzanie animacji zostało skorelowane z dynamiką aktywności ruchowej publiczności, również dźwięk zamierzał powiązać z interakcją. Ponieważ znaleźliśmy się już wcześniej i Kostia wiedział, że zajmuję się mediami interaktywnymi zaproponował mi udział w tworzeniu interaktywnego komponentu instalacji. Rozmawiając i proponując sobie rozmaite pomysły szybko doszliśmy do koncepcji budowy miejsca, w którym skanowana przez sensory fizyczna aktywność odwiedzających jest transformowana w zmiany elektronicznie generowanego środowiska (dźwięk i projekcje) – między innymi stąd taki a nie inny tytuł instalacji. Równocześnie pewne elementy tego środowiska (wspomniana kolumna) wzięte zostały z przestrzeni fizycznej, tak więc zamiarem naszym było stworzenie pewnej rzeczywistości mieszananej, układu wzajemnie się uzupełniających elementów fizycznych i wirtualnych.

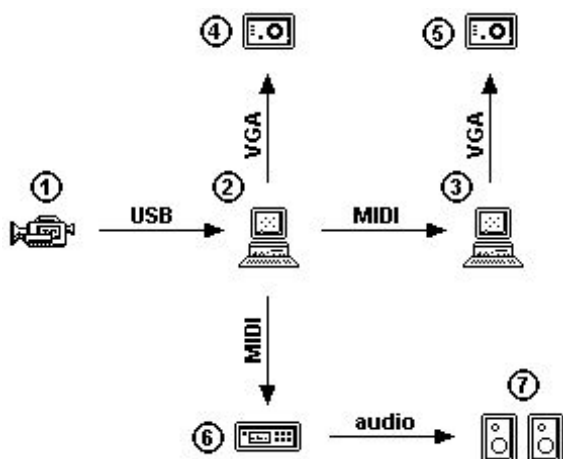


*Kolumna będąca głównym tematem animacji: od fizycznego pierwowzoru do modelu 3D.*

Omówiliśmy różne warianty realizacji instalacji biorąc pod uwagę możliwości czasowe, technologiczne i specyfikę lokalizacji instalacji. Od początku spodziewaliśmy się problemów związanych z niewielkimi wymiarami pomieszczenia ekspozycyjnego. Nie pomyliliśmy się – problemy te zostaną opisane poniżej.

Osobną i w zasadzie podstawową kwestię stanowiło oprogramowanie dla instalacji. Musieliśmy skorelować dane o aktywności publiczności (a co za tym idzie wybrać metodę pobierania tych danych) z wyświetlanym wideo i dźwiękiem – ze względu na użycie dwóch komputerów trzeba też było wybrać najskuteczniejszą metodę komunikacji pomiędzy uruchomionymi na nich aplikacjami.

## INTERAKCJA

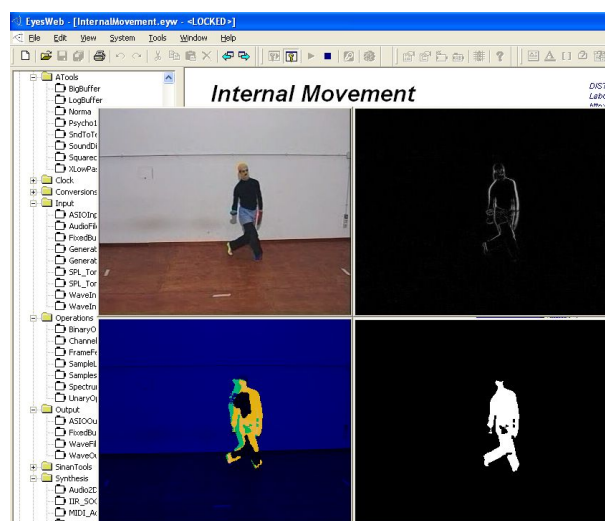


Kamera USB (1) przesyła obraz do komputera z uruchomionym programem EyesWeb (2), który steruje przez interfejs MIDI drugim komputerem z główną animacją (3, 5), własnym projektorem (4), modulem MIDI (6, 7).

Narzucającą się w naturalny sposób metodą organizacji interakcji dla podsystemu wideo było zbieranie przez 3 sekundy (przypomnę, że tyle właśnie trwała pojedyncza sekwencja animacji) informacji o aktywności publiczności, następnie na tej podstawie wybranie kolejnej sekwencji, itd. Podsystem dźwiękowy natomiast z założenia miał pracować w pełnym czasie rzeczywistym, reagując natychmiast na aktywność publiczności (początkową koncepcję przyporządkowania każdej z pętli wideo jakiegoś konkretnego dźwięku odrzuciliśmy jako nieciekawą – zamiast tego postanowiliśmy zbudować generatywną kompozycję, w której charakter muzycznych struktur będzie uzależniony od sygnałów napływających z systemu śledzenia ruchu).

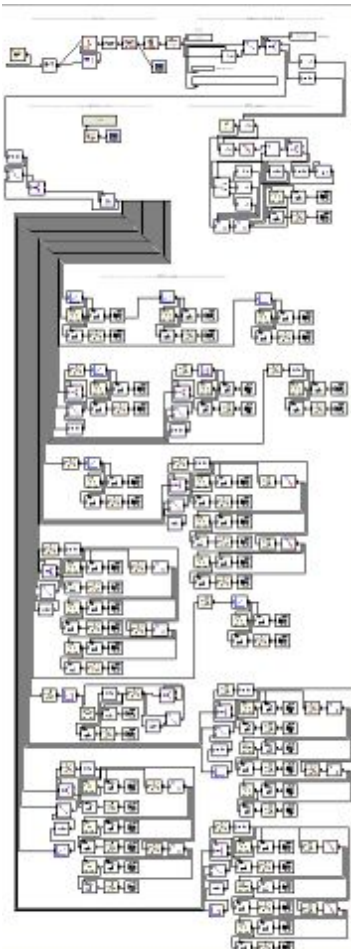
Ważnym szczegółem było przystosowanie oprogramowania do pracy w zmiennych warunkach interakcji (z jedną lub wieloma osobami wizytującymi) – oznaczało to, że oprogramowanie powinno mieć zdolność ograniczania “czułości” instalacji na aktywność publiczności w sytuacji, gdy aktywność ta będzie przez dłuższy czas wysoka (wiele osób jednocześnie w skanowanej przestrzeni) i odwrotnie: zwiększania reaktywności instalacji w przypadku długotrwałej niskiej aktywności, tak by jedna osoba również była w stanie wywoływać zmiany w funkcjonowaniu instalacji.

Podstawową aplikacją, czy raczej środowiskiem pracy, którego zamierzałem użyć do opracowania aplikacji zarządzającej interakcją jest EyesWeb – program, który zwykle stosuję jako główne narzędzie podczas pracy nad instalacjami interaktywnymi, ale również nad innymi projektami wykorzystującymi oprogramowanie. Program ten należy do bogatej rodziny software'u obsługującego pracę z mediami w czasie rzeczywistym: digitalizowanym wideo, dźwiękiem, MIDI, danymi liczbowymi i tekstowymi. Posiada spory zestaw gotowych komponentów służących do przetwarzania mediów mieszanych i reżyserowania interakcji. Umożliwia też łatwe opracowywanie własnych rozszerzeń, dzięki czemu można uzupełnić go w razie potrzeby o brakujące elementy. Ważną cechą programu jest modułarna architektura, niegdyś popularna w sprzętowych syntezatorach muzycznych a obecnie przeżywająca swój renesans głównie dzięki oprogramowaniu (np. popularnemu instrumentowi wirtualnemu Native Instruments Reaktor).



EyesWeb: uruchomiony jeden z przykładowych patchy ilustrujący możliwości przetwarzania wideo w czasie rzeczywistym.





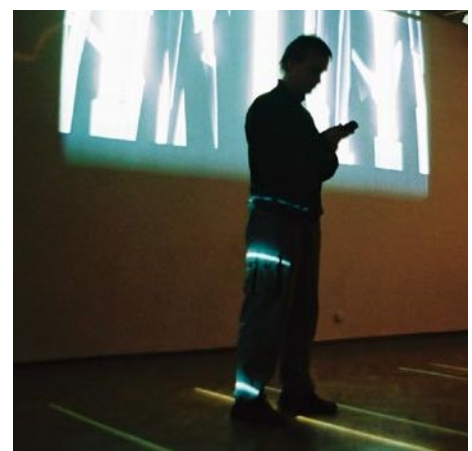
*EyesWeb: schemat aplikacji (patcha) zarządzającego śledzeniem ruchu i wysyłającego komunikaty MIDI do modułu brzmieniowego i komputera wyświetlającego główną animację.*

EyesWeb, a dokładniej zbiór programowych modułów obsługujących motion capture, przyjmował i analizował dane z kamery, wykrywając różnice pomiędzy kolejnymi klatkami obrazu – nie trzeba wyjaśniać, że to właśnie te różnice świadczą o aktywności publiczności. Motion capture albo motion tracking to ogólne określenia systemów śledzących lub wykrywających ruch. Wywodzą się one przede wszystkim z technologii stosowanych w systemach bezpieczeństwa. Zazwyczaj sensorem wykorzystywanym w tego rodzaju aplikacjach jest jakiś rodzaj kamery: może być to zwykła kamera rejestrująca światło widzialne, kamera podczerwieni czy termiczna. W przypadku naszej instalacji użyliśmy prostej kamery “internetowej” na złączu USB.

Zastosowanie kamery działającej w zakresie światła widzialnego implikowało oczywiście oświetlenie takim światłem skanowanej powierzchni (czyli mówiąc banalnie: podłogi) – wymóg stojący w jaskrawej sprzeczności z koniecznością wyciemnienia pomieszczenia w związku z rzucanymi na ścianę projekcjami. Aby dostarczyć kamerze światła a jednocześnie nie rozjaśnić pomieszczenia użyliśmy projektora wideo rzucającego na podłogę animację w formie przesuwających się równoległe do ściany z główną projekcją pasów światła. Dawało to ciekawy, nawiązujący do idei i tytułu instalacji efekt wizualny, jednocześnie rozwiązując problem oświetlenia. Równocześnie zainspirowało Kostię do dodatkowego naklejenia na podłogę pasów białej taśmy – fizycznych odpowiedników pasów światła. Musieliśmy jedynie poeksperymentować z ustawieniem kamery w taki sposób, by najlepiej wykorzystywała odbite światło, oraz przystosować oprogramowanie do pewnego stałego poziomu różnic pomiędzy kolejnymi klatkami obrazu z kamery (innymi słowy: przesuwające się pasy światła i niewielkie przypadkowe zmiany oświetlenia nie powinny wyzwalać impulsów, dopiero zaburzenie tego stałego poziomu “szumu tła”, świadczące o aktywności publiczności miało uruchamiać reakcje instalacji).

Istniejące systemy motion capture, w tym również zintegrowany z EyesWeb potrafią dość precyzyjnie lokalizować obiekty poruszające się w polu widzenia kamery, jednak by ową precyzję osiągnąć należy spełnić kilka warunków, z których najważniejszym jest możliwie duża odległość kamery od lokalizowanego obiektu. Ten dziwny na pierwszy rzut oka wymóg bierze się z potrzeby zredukowania wpływu odległości obserwowanego obiektu od osi obiektywu na jego obraz widziany przez kamerę.

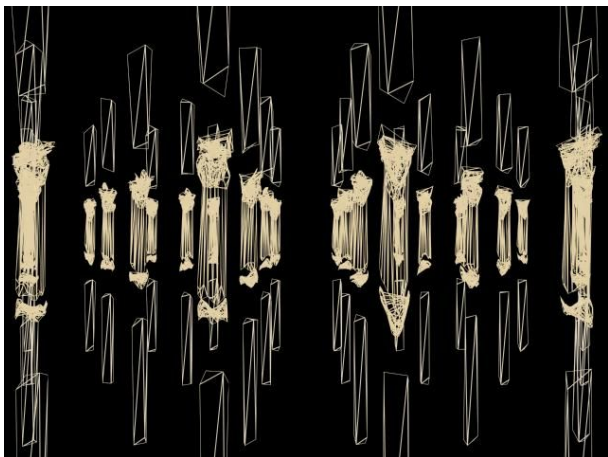
Niestety przestrzeń Entropii nie dawały nam możliwości umieszczenia kamery w wystarczającej odległości, dlatego zdecydowaliśmy się na rozwiązanie kompromisowe: kamerę umieściliśmy pod kątem (nie



*Ruchome pasy białego światła pozwalały “widzieć” kamerze.*

nad skanowaną powierzchnią), oprogramowanie badało ogólną aktywność i jej dynamikę w skanowanym polu – nie przejmowaliśmy się natomiast dokładną lokalizacją osób przemieszczających się w polu widzenia kamery.



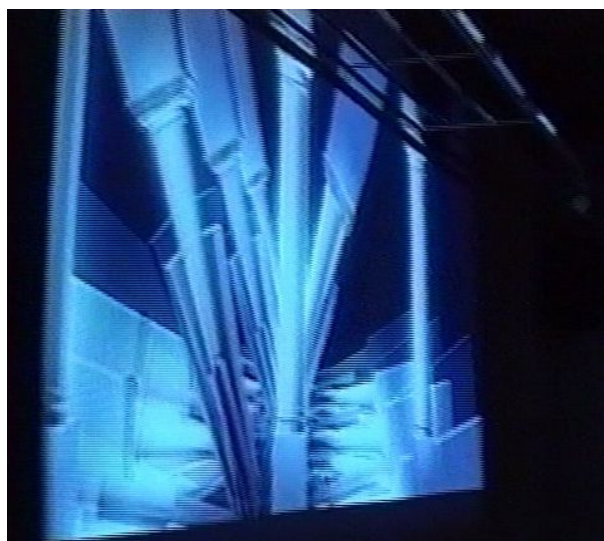


*Jeden z projektów animacji wykorzystanych w SKANERze.*

Specyficznym problemem, z którym musiał zmierzyć się Kostia, a który był również związany z niewielkimi wymiarami pomieszczenia, dotyczył projekcji. Aby uzyskać obraz o odpowiedniej wielkości z projektora wideo należy go oczywiście umieścić w wymaganej przez konstrukcję projektora (a dokładniej obiektywu) odległości od ekranu. Tego dystansu wyraźnie nam brakowało, Kostia poradził sobie z kłopotliwą sytuacją za pomocą lustra odbijającego światło projektora. Osoby znające inne realizacje Kostii zdają sobie zapewne sprawę, że tego rodzaju manipulacje z odbijaniem światła projektora od jednej lub wielu płaszczyzn są

charakterystyczną techniką pracy: przy pomocy symulacji w 3D Studio Max Kostia ustalała wzajemne położenie projektorów, lusterek i ekranów by uzyskać odpowiedni efekt powiększenia obrazu bez jego zniekształcenia (o ile oczywiście nie jest to jego zamierzeniem).

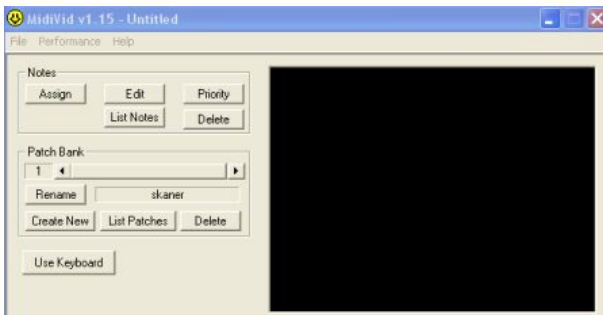
Od początku najwięcej problemów związanych było z wyzwaniem głównych sekwencji wideo: o ile praca z np. jednym plikiem wideo odtwarzanym w pętli nie nastręcza problemów, o tyle kilkanaście czy kilkadziesiąt krótkich pętli wideo w rozdzielczości 800x600 pikseli i prędkości 25 fps wyzwalały w zmiennej kolejności tak, by odtwarzały się płynnie i bez przerw stanowi już wyzwanie dla większości programów odtwarzających. Większość z nich potrzebuje trochę czasu by utworzyć potrzebne bufora w pamięci RAM, przez co każdorazowa zmiana odtwarzanego pliku na inny związana jest z krótkim, lecz zauważalnym zatrzymaniem odtwarzania – ma to znaczenie zwłaszcza w przypadku pracy z wideo o wysokiej rozdzielczości.



*Główna projekcja rzutowana na jedną ze ścian galerii.*

Używając EyesWeba w przypadku operowania na większej ilości wysokorozdzielczych klipów dobrze jest umieścić je wszystkie w jednym pliku avi lub mpeg. EyesWeb pozwala łatwo przeskakiwać do wybranej klatki obrazu lub odtwarzać fragment pliku (określony przez klatkę początkową i końcową) w pętli. Niemniej podczas testów stwierdziliśmy, że płynność odtwarzania obrazu nie zadowalała nas. Z drugiej strony: zależało nam tylko na płynnym odtwarzaniu animacji i możliwości łatwego sterowania odtwarzaniem z poziomu zewnętrznej aplikacji, więc użycie EyesWeba nie było konieczne – mogliśmy zastąpić go jakimkolwiek programem spełniającym opisane wyżej wymagania. Początkowo rozważałem możliwość napisania własnego odtwarzacza wideo z portem OSC (dla współpracy z Eyw), jednak przeglądając recenzje oprogramowania w serwisie

<http://vjcentral.com> natrafiłem na prosty program MidiVid, który, jak się okazało, doskonale nadawał się do wykorzystania w instalacji.

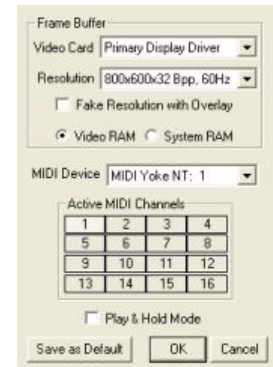


*MidiVid: główne okno programu w chwili po uruchomieniu.*

MidiVid jest niewielkim, prostym w obsłudze i niezwykle skutecznym programem służącym do wyzwalania sekwencji wideo za pośrednictwem komunikatów MIDI. Pozwala zbudować własny bank sekwencji (przyporządkowując poszczególnym nutom pliki wideo) wideo i sterować ich odtwarzaniem np. poprzez klawiaturę MIDI lub zewnętrzny

sequencer. Procedury odtwarzające wideo są tak skonstruowane, by odtwarzanie sekwencji rozpoczynało się natychmiast po przesłaniu komunikatu (potrzebne bufory są wcześniej tworzone w pamięci komputera lub karty graficznej).

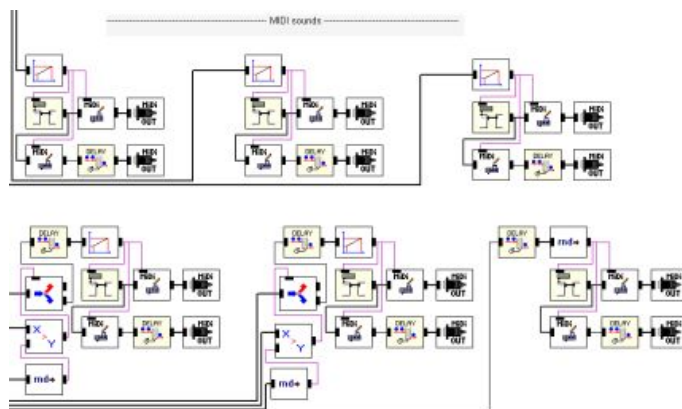
Tak więc co 3 sekundy EyesWeb wysłał przez interface MIDI komunikat do komputera, na którym uruchomiony był MidiVid, a ten odtwarzał wybrany plik wideo. Metoda była prosta i skuteczna. Połączenie komputerów przez interface'y MIDI okazało się o wygodniejsze od Ethernetu i do naszych celów zupełnie wystarczające.



*Okno ustawień parametrów wyświetlania obrazu i obsługi MIDI.*

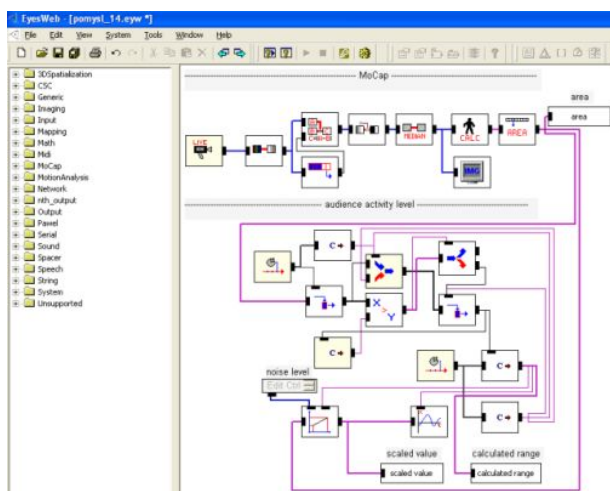
Dźwięk w "SKANERze" początkowo planowałem zbudować w oparciu o CyberSound lub Pure-Data. Oba systemy pozwalają na pracę z interaktywną muzyką: komponowanie, tworzenie brzmień i aranżacji. Niestety budżet instalacji nie pozwalał nam na użycie jeszcze jednego szybkiego komputera, tak więc wykorzystałem jeden z instrumentów z mojego studia: leciwy ale ciągle dobrze brzmiący moduł brzmieniowy Yamaha MU50 (rozwiązanie to miało też tę zaletę, że rozmiary MU50 są daleko mniejsze niż najmniejszego laptopa, nie mówiąc już o stacjonarnym komputerze, a nie chcieliśmy przesadzić z gromadzeniem sprzętu) sterowany przez MIDI (komunikaty MIDI generował i wysyłał EyesWeb).

Podczas przygotowywania instalacji dyskutowaliśmy z Kostią problemy związane z interakcją. Użycie przez nas 3 sekundywch sekwencji wideo w dość istotny sposób ograniczało możliwości interakcji, założyliśmy więc, że muzyka w naszej instalacji będzie reagowała na poczynania publiczności w wyraźny sposób – bardziej bezpośrednio w porównaniu z charakterem zmian w materiale wideo. Muzyczny komponent instalacji zyskiwał na znaczeniu w miarę rozmów i prac nad całością. W pewnym momencie zaczęliśmy traktować SKANER jako wirtualny instrument audiowizualny.



*EyesWeb: fragment patcha sterującego modulem brzmieniowym.*

MU50 nie daje zbyt wielkiego pola do popisu dla kreatorów brzmień, jednak przy odrobinie wysiłku można wydobyć z niego naprawdę interesujące dźwięki, zaś 16 częściowy multitimbral przy polifonii równej 32 był dla nas zupełnie wystarczający. Korzystając z możliwości EyesWeb zaprogramowałem 16 różnych struktur dźwiękowych wyzwalanych przez sygnały z motion capture – każda odpowiadająca jednej z części multitimbral modułu. Różne parametry owych struktur były modulowane sygnałami z motion capture. Każda ze struktur posiadała pewne swoje cechy i w różny sposób reagowała na zmiany zachodzące w instalacji: niektóre wyzwały pojedyncze tony, inne ich ciągi (których długość mogła zależeć np. od średniej aktywności w dłuższym czasie), bądź wielodźwięki, część struktur reagowała na sygnały z pewnym opóźnieniem. Wszystkie te generatywne struktury produkowały komunikaty MIDI wysyłane w czasie rzeczywistym do MU50. W rezultacie poruszający się widzowie wywoływali ciągi muzycznych sekwencji rozwijających się w razie ciągłej aktywności i powoli rozrzedzających w momentach zaniku tej aktywności.



*EyesWeb: główne okno programu (widoczny fragment patcha użytego w SKANERze).*

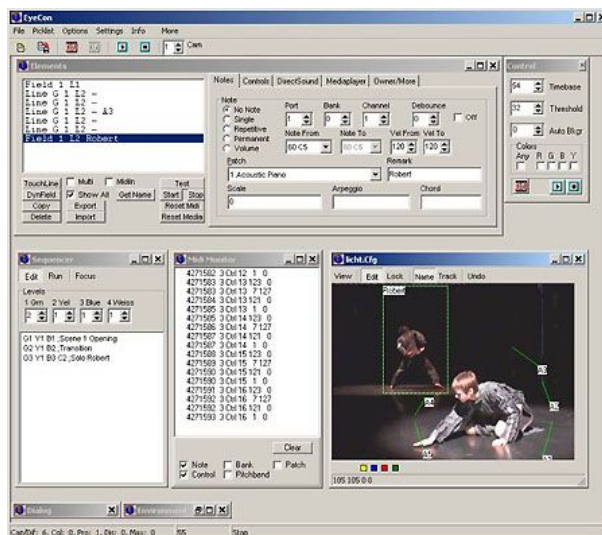
Użyty w SKANERze EyesWeb jest darmową aplikacją dostępną na platformę PC/Win opracowaną przez InfoMus Lab (Laboratorio di Informatica Musicale / DIST - University of Genova) jako część prowadzonego przez UE projektu MEGA (Multisensory Expressive Gesture Applications) związanego z sensorycznym rozpoznawaniem ruchu i gestów. Posiada pewne ideowe związki z opisanym niżej EyeCon (jeden i drugi program dysponuje świetnymi procedurami śledzenia ruchu, więc nadają się do wykorzystania wszędzie tam, gdzie zachodzi konieczność analizy danych wizualnych pod kątem wychwycenia informacji o aktywności ruchowej), niemniej różni się interfejsem

(EyesWeb jest systemem modułowym) i bardziej otwartą architekturą. Z całą pewnością nie jest programem zapewniającym tak spektakularne efekty wizualne jak CSoundAV, czy VVVV (również opisane niżej) w przypadku wykorzystania go jako narzędzia do pracy z wideo w czasie rzeczywistym, podobnie ma się rzecz z dźwiękiem, tak więc VJ-je i osoby zainteresowane real-time image processingiem mogą być rozczarowane EyesWeb (w takim przypadku polecam lekturę materiałów z witryny <http://vjcentral.com> grupującej odnośniki i opisy programów tego rodzaju) Niemniej doskonała dokumentacja i tzw. SDK (Software Development Kit), czyli komponenty pozwalające na samodzielne programowanie rozszerzeń dla aplikacji (wymagana jest do tego znajomość C++) pozwalają wykorzystać EyesWeb jako narzędzie pomocne zwłaszcza w pracach eksperymentalnych.

URL <http://www.eyesweb.org>

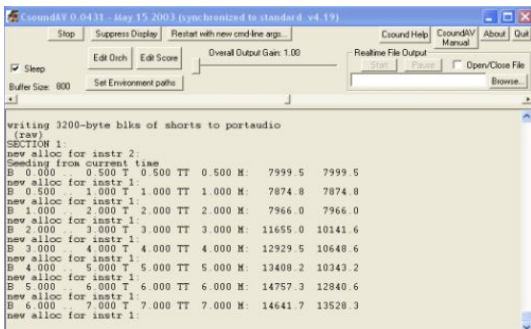
EyeCon jest dość specyficznym programem. Jest to aplikacja komercyjna (choć mówi się o udostępnieniu jej w bezpłatnej formie) dostępna na platformę PC/Win. Interfejs EyeCon oparty jest o listwę czasu, co może zachęcać do użycia go osoby, które posługiwały się wcześniej np. Flashem czy Directorem lub jakimkolwiek programem do edycji wideo albo sequencerem muzycznym. EyeCon powstał przy aktywnym udziale kompanii teatralnej Palindrome, dzięki czemu szczególnie nadaje się do wykorzystania w działaniach teatralnych i parateatralnych wykorzystujących media elektroniczne, zaś doświadczenie autora w budowie m.in. przemysłowych systemów bezpieczeństwa zaowocowało doskonałym modulem detekcji ruchu (motion tracking).

URL <http://eyecon.palindrome.de>



*Interfejs programu EyeCon z widocznym oknem systemu śledzenia ruchu.*





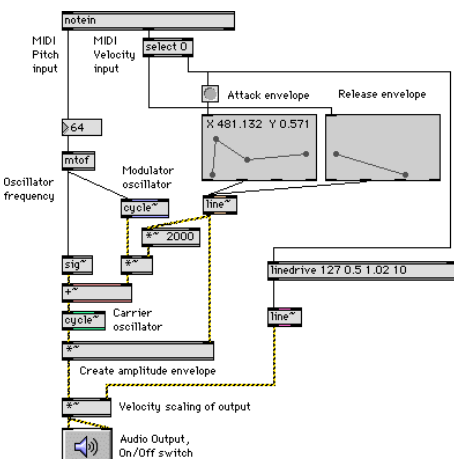
CsoundAV: główne okno programu.

kompozytora i programisty Gabriela Maldonado).

Csound jest swego rodzaju przemysłowym i laboratoryjnym standardem opisu algorytmów generowania i przetwarzania dźwięku – wiele nowych technik syntezy czy przekształcania dźwięku znajduje swoje pierwsze implementacje właśnie w Csound, zanim zostaną – lub nie – wykorzystane w komercyjnych aplikacjach i sprzęcie.

Csound jest implementowany na wszystkich ważnych platformach sprzętowych i software'owych. W przeciwieństwie do innych opisanych tu programów Csound jest językiem programowania, nie graficznym środowiskiem (jakkolwiek istnieją graficzne nakładki i rozmaite programy ułatwiające pracę z Csound), więc może być kłopotliwy dla wielu użytkowników. Warto jednak pamiętać, że Csound jest programem o bodaj największych możliwościach generowania i obróbki dźwięku oraz wideo i animacji 3D w kontekście ich związków z dźwiękiem. Również mnogość implementacji (w tym jako wtyczki VST, PD, MAX/MSP) nie pozostaje bez znaczenia.

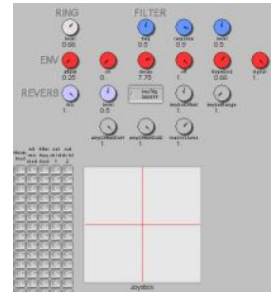
URL <http://csounds.com>



MAX/MSP: przejrzysty graficzny interface pozwala łatwo budować skomplikowane struktury dźwiękowe.

URL <http://www.cycling74.com>

CyberSound (lub CSound) jest językiem programowania opracowanym w MIT i skoncentrowanym początkowo na obróbce dźwięku (ze względu na zaawansowane możliwości i wysoką konsumpcję mocy obliczeniowej początkowo nie był to system czasu rzeczywistego, którym jest obecnie), obecnie uzupełniany o elementy graficznego interface'u użytkownika i możliwości pracy z wideo i animacją 3D (CsoundAV autorstwa

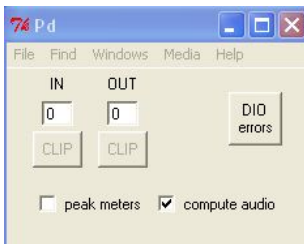


Csound umożliwia tworzenie własnych graficznych interface'ów...



... oraz łączenie dźwięku z wideo i animacją 3D.

MAX/MSP (i jego inne wersje) to w zasadzie pierwszy popularny medialny system czasu rzeczywistego pracujący jako środowisko w którym łączymy ze sobą niezależne moduły. Opracowany w IRCAM, podobnie jak Csound początkowo był aplikacją wyłącznie dźwiękową, ale obecnie umożliwia już integrowanie różnych mediów. MAX jest programem komercyjnym (jakkolwiek istnieją również – choć są mniej popularne – wersje niekomercyjne, jak np. j-MAX dla Linuxa), do niedawna dostępnym tylko na platformę MacOS, obecnie również na PC/WIN. MAX jest programem niezwykle popularnym, na co złożyło się kilka czynników: oprócz ewidentnej "pionierskości" programu (w chwili swej premiery MAX właściwie nie miał konkurencji) na jego sukces wpłynął czytelny i wydajny interface.

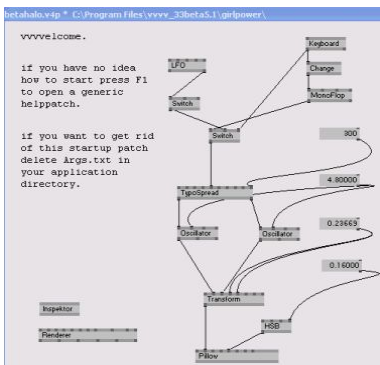


Pure-Data: niepozorne główne okno programu w chwilę po starcie.

Pure-Data (PD) można uznać za następcę MAXa (autorem programu jest Miller Smith Puckette, który stworzył również pierwsze wersje MAXa). W odróżnieniu od swego poprzednika jest aplikacją bezpłatną i dostarczaną wraz z kodem źródłowym, oraz rozwijaną na licencji open-source przez międzynarodowy zespół developerów-wolontariuszy. PD jest wieloplatformowy, istnieją skompilowane wersje nie tylko dla Windows, Linuxa, czy MacOS, ale też wersje dla systemów palmtopowych.

Pure-Data już w tej chwili dysponuje potężnymi możliwościami, a przy tym rozwija się bardzo dynamicznie – z całą pewnością jest to oprogramowanie “z przyszłością”, zaś częściowa kompatybilność z MAX/MSP jest również wielką zaletą. PD doczekał się też doskonałego zestawu wtyczek do pracy z mediami wizualnymi (GEM), nie gorszymi od NATO dla MAX/MSP.

URL <http://Pure-Data.org>

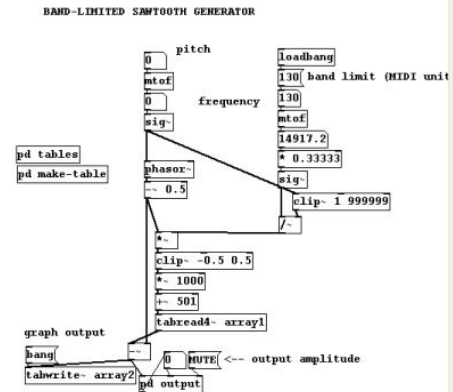


VVVV: typowy modułowy interface...

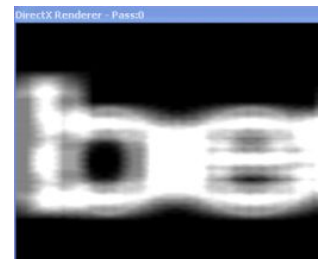
Program VVVV firmy Meso jest modułowym syntezatorem wideo: ciągle w wersji beta, jednak posiada potężne możliwości pracy z animacją komputerową w czasie rzeczywistym. Szczególnie unikalne możliwości w zakresie modelowania animacji 3D stawiają VVVV w rzędzie najlepszych aplikacji tego rodzaju. Autorzy programu używali go z powodzeniem do prac tak różnych jak przygotowanie video-clipów, VJ-ing i wizualizacje architektoniczne, poprzez różnego rodzaju prezentacje, aplikacje i instalacje interaktywne.

URL <http://vvv.meso.net>

Powyższe zestawienie nie wyczerpuje, rzecz jasna, tematu oprogramowania wspomagającego pracę z mediami czasu rzeczywistego (pomiędzy innymi takie programy jak OpenMusic, SuperCollider, Imagine, Isadora i wiele innych). Opisane środowiska niekoniecznie mogą okazać się najlepszym wyborem w konkretnej sytuacji – niekoniecznie muszą też być najłatwiejsze do nauczenia się. Natomiast z całą pewnością zapoznanie się z którymkolwiek z nich pomoże w pracy z innymi podobnymi programami.



Kilka modułów połączonych w funkcjonalną całość.



...i olbrzymie możliwości manipulowania obrazem dzięki DirectX



**sprzęt:**

- projektor wideo x 2
- komputer pc x2
- kamera internetowa USB
- Yamaha MU50
- interface MIDI x 2
- nagłośnienie

**oprogramowanie:**

- 3D Studio Max
- EyesWeb
- MidiVid

## AUTORZY



autorzy

SKANER – instalacja interaktywna

**Bartosz “Kostia” Jakubicki** – idea, animacja 3D, aranżacja przestrzeni

**Paweł Janicki** – programowanie, muzyka

### **Paweł Janicki**

[pawel@wrocenter.pl](mailto:pawel@wrocenter.pl)

<http://ping.wrocenter.pl>

<http://gameboyzz.terra.pl>

inne projekty:

<http://generator.wrocenter.pl>

<http://pod.wrocenter.pl>

Twórca medialny i producent. Pracuje z muzyką generatywną, systemami interaktywnymi i własnym, stale rozwijanym hardware i software. Współpracuje z WRO Centrum Sztuki Mediów i wieloma innymi instytucjami i artystami. Członek Gameboyzz Orchestra Project.

*Media artist and producer. Working with generative music, interactive systems and own constantly developed software/hardware. Co-operates with WRO Center for Media Art, and other various artists and institutions. Member of Gameboyzz Orchestra Project.*

### **Bartosz “Kostia” Jakubicki**

[kostia@box43.pl](mailto:kostia@box43.pl)

Artysta, designer, architekt wnętrz. Studiował w Akademii Sztuk Pięknych we Wrocławiu, jest związany z tą uczelnią pracując jako asystent na Wydziale Architektury Wnętrz. Zajmuje się badaniami relacji pomiędzy przestrzeniami wirtualnymi i fizycznymi, zwłaszcza w kontekście architektury. Pracuje także jako scenograf i architekt wnętrz, oraz jako muzyk i VJ w grupie Syntax Error.

*Artist, interior designer. Studied on The Academy of Fine Art at Wrocław, and working at this school as assistant on Interior Design Department. He's main research interest is relationship between virtual space and real interiors, especially in architectural context. Also working as scenography and interior designer, and musician/VJ in his group Syntax Error.*

## PODZIĘKOWANIA

Autorzy instalacji pragną podziękować kuratorom Galerii Entropia i organizatorom Wrocław Industrial Festival za przychylność i pomoc w realizacji instalacji.

tekst (na zlecenie Syntezatory.pl): **Paweł Janicki**  
współpraca: **Bartosz “Kostia” Jakubicki**